MODELING OF PRIMARY REFORMER TUBE METAL TEMPERATURE (TMT)

MUHAMMAD ASRIF BIN ABDUL RAHIM

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

Modeling of Primary Reformer Tube Metal Temperature (TMT)

By

MUHAMMAD ASRAF BIN ABDUL RAHIM

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

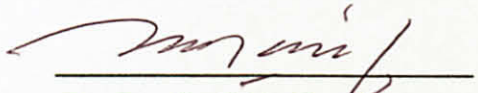
Modeling of Primary Reformer Tube Metal Temperature (TMT)

by

Muhammad Asraf bin Abdul Rahim

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



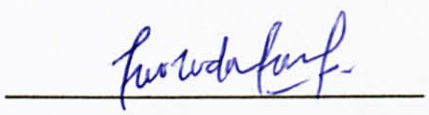
(Dr. Rosdiazli bin Ibrahim)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



(Muhammad Asraf bin Abdul Rahim)

ABSTRACT

This report serves to give the details of the development of the project and the steps taken in realising the Modeling of Primary Reformer Tube Metal Temperature (TMT). The main aim in this project is to develop an adequate model to predict primary reformer TMT based on real-time data obtained from PETRONAS Ammonia Sdn. Bhd. (PASB) plant. The model being developed shall serve to give adequate prediction of future outputs which in this case are temperature of the reformer tubes, given the process variables which act as inputs. The significance of this project are to initiate the creation of a robust predictive control to prevent overheating from occurring as well as to help the plant operator to plan the Preventative Maintenance if any of the unavoidable situation causing overheating occurs such as feedstock/steam failure, restricted process flow or the burners misalignment [1]. The tube temperature can also be optimised to yield better production while generating more profits. This report outlines the heuristic approaches taken in modeling the TMT using Artificial Neural Network (ANN), and other black box (empirical) models developed using system identification (AutoRegressive Exogenous) and Multiple Linear Regression (MLR) meant to serve as benchmarks. Literature surveys on the general idea of reformer, ammonia production, failures of the reformer tubes and the modeling related to dynamic systems have been conducted and discussed throughout this report.

ACKNOWLEDGEMENT

First and foremost, the author would like to express his utmost gratitude to his supervisor Dr. Rosdiazli b. Ibrahim for all his never ending support and his contribution to this project. The author also would like to express his heartfelt appreciation to Dr. Rosdiazli for all the time he spent and all his guidance, motivation and encouragement that he provides to author throughout the entire project stint.

Many thanks to Vishal Nanji Patel for the early works done involving the literature review, the early processing of the data as well as the modeling of primary reformer TMT using Artificial Neural Network.

The author would like to expresses his greaest appreciation to Ms. Maryam Jamela bt. Ismail for her willingness to share her knowledge in completing this project. Thank you also for Ms. Siti Hawa for always being such a helping hand in giving advices regarding the report format and important datelines.

Lastly, the author would like to thank his family; especially his parents for always pouring their never ending love and moral supports in contributing to the completion of this Final Year Project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

TMT	Tube Metal Temperature
LOTIS	Laser optical tube inspection system
PASB	PETRONAS Ammonia Sdn. Bhd.
OPU	Operating Unit
RBF	Radial Basis Function
GTS	Global Technology Solution
MSE	Mean Square Error
MAE	Mean Absolute Error
SSE	Sum Square Error
MSEREG	Mean Square Error with Regularization
RMSE	Root Mean Squared Error
DCS	Distributed Control System
PV	Process Variable
ANN	Artificial Neural Network
MLP	Multi Layer Perceptron
MLR	Multiple Linear Regression
ARX	AutoRegressive eXogenous
GA	Genetic Algorithms
MISO	Multi Input Single Output
GUI	Graphical User Interface

CHAPTER 1
INTRODUCTION

1.1 Background Study

Primary reformer is the largest and the most expensive item used in an ammonia, methanol, or hydrogen plants. Steam reforming is the most widespread process and the economical routes for the production of hydrogen-rich synthesis gas from light hydrocarbon. The feed materials are naphtha, natural gas, or liquid gas are to be heated up and distributed into the catalyst-filled reformer tubes (arranged in vertical rows) and react with steam to produce CO, CO_2, and H_2 under high temperature and pressure [2,3].

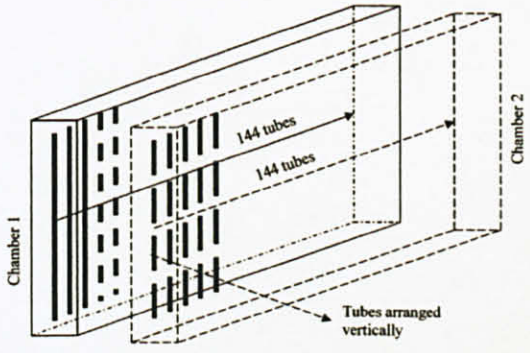


Figure 1 : Tubes Arranged Vertically in PASB's Primary Reformer

To avoid failure of reformers to occur in plant, technological improvements and good operation of the entire plant are both critically needed. Two factors are mainly affecting the failure mechanism of tubes which are the thermal stresses of the tube wall and by the stress imposed of the operation under pressure. The temperature of the tube metal (also called tube wall) influences the tube life as when overheating occurs; it causes dramatic reduction in the tube life. The rule of thumb is that a rise in operating temperature of only 20°C above designated temperature limit will approximately halve the remaining operating life of a tube [3]. Thermal cycling (heated and cooled down alternately) can also lead

to the acceleration of creep damage. The creep strength (the resistance from the creep damage) of a tube is also influenced by the tube design, tube material, and the catalyst effects apart from the tube wall temperature earlier stated. The tubes failures are to be avoided as it is important to minimise cost and to avoid unscheduled plant shut-downs that could result to losses.

1.2 Problem Statement

Overheating of reformer tubes can cause dramatic reduction in the tube life. There is a need to develop an empirical model that can adequately predict the primary reformer TMT and is due to several factors.

Firstly, there is currently no online monitoring system to directly monitor temperature of the primary reformer tubes in real time. This is due to the high operating temperature (about 800-1000°C) as well as expensive cost. Current practice done by operators in oil and gas industries in order to prevent the tube failure is by measuring the outer TMT using pyrometer through peepholes manually. However, this imposes problems regarding the personnel health and safety due to the high ambient temperature that reaches 40°C and might also lead to the human error that affects the accuracy [4].

The second problem, currently there is currently no model prediction for planned maintenance (preventive maintenance) for primary reformer tubes. The primary reformer tubes condition only will be checked if there are maintenance shut-downs (breakdown maintenance). Certain other nondestructive testing (NDT) methods being used are; ultrasonic attenuation, eddy current inspection, Metallography, diameter measurement, and the laser optical tube inspection system (LOTIS) [3]. However, these methods are just to check the tubes' conditions and not act as the precautionary acts to prevent tube damages.

The other common precautionary act to prevent tubes failure is to operate the primary reformer below the designed limit temperature, but the issue of using such method is the production of a plant cannot be optimised.

1.3 Objectives

The objectives of this project are:

- To identify critical process variables (inputs) in predicting the primary reformer TMT (output)
- To develop Artificial Neural Network (ANN) model that can adequately predict the primary reformer TMT
- To develop other alternative models using AutoRegressive Exogenous (ARX) and multiple linear regression (MLR) as benchmarks

1.4 Scope of Study

The scope of study involves:

- Failure analysis of the steam reformer tubes
- Performing statistical analysis in identifying the critical process variables
- Proposing possible empirical models (black box) to predict primary reformer TMT (offline training –batch mode)
- Performing heuristic approaches in obtaining optimal ANN structure
- Developing and comparing performance of other empirical models with ANN

The models are being developed by using MATLAB software. The significances of this project are, in preventing tube failures, minimising cost, avoiding unscheduled plant shut down that leads to huge losses, and optimising the tube temperature to yield better production while generating more profits.

This study perhaps can initiate the effort to create a robust monitoring system or predictive control strategy (e.g. Internal Model Control, Model Predictive Control) that can be integrated into an existing control system in order to safeguard the related process equipments.

CHAPTER 2
LITERATURE REVIEW/THEORY

2.1 Primary Reformer in Ammonia Plant

In modeling the primary reformer TMT, PASB's plant will be used as a pilot plant. Real-time data was collected from PASB to be used in developing the empirical models.

Figure 2 :Primary Reformer(left) and Secondary Reformer(right)[5]

There are four furnace types (firing configurations) currently in used today:

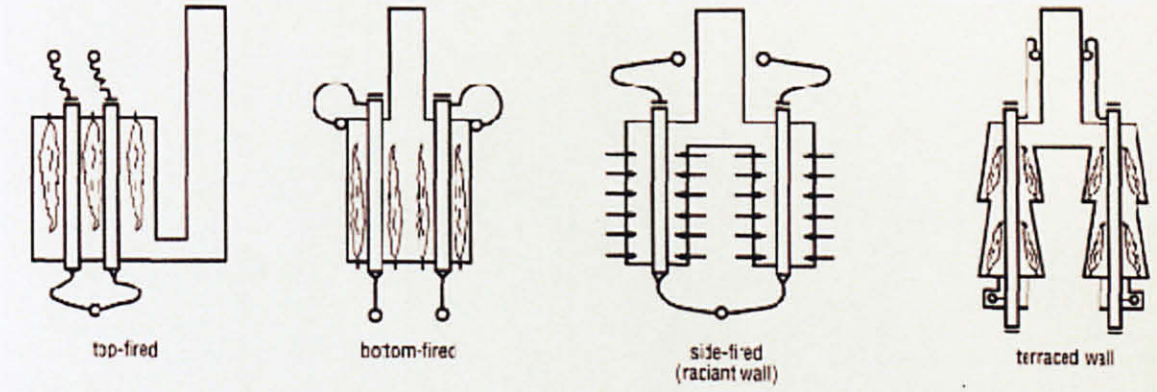


Figure 3 :Main Types of Steam Reforming Furnace [3]

Below picture depicts the layout of the primary reformer used for steam reforming in PASB plant which is the side-fired configuration:

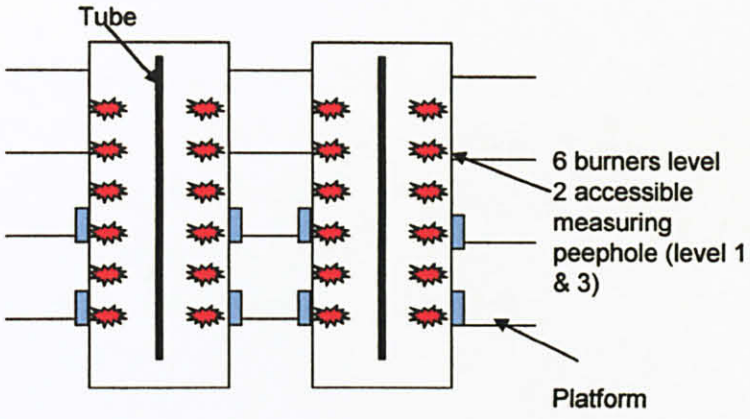


Figure 4 : Frontal View of PASB's Primary Reformer [5]

The PASB plant is using the side-fired (radiant wall) reformer with 6 levels of burners, and total up to 288 tubes (2 chambers, 144 tubes each). Temperature of products combustion in the furnace chamber varies with the furnace configuration and dimensions, reformer tube arrangements, and the position and number of burners [4]. At each level, there are 18 peep holes on each side of the wall.

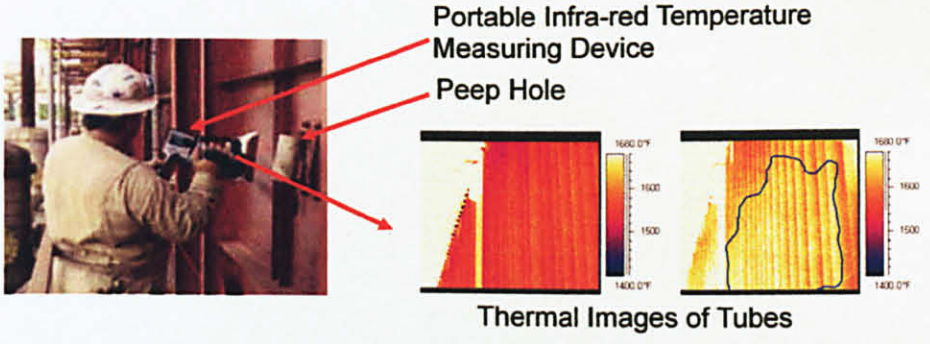


Figure 5 : Manual Measurement Using Pyrometer Through Peepholes [5]

These peep holes are used to measure the tube temperature by using pyrometer (portable infra-red temperature measuring device). Only level 1 and

level 3 are accessible for the operator to manually measure the temperature of tubes' surfaces.

2.2 The Ammonia Production

Nitrogen based fertilizers use Ammonia as the basic material for providing the nitrogen component. Ammonia is synthesised by chemically combining hydrogen and nitrogen, in the presence of a catalyst, commonly nickel. The hydrogen requirement is met by reacting a mixture of steam and hydrocarbons, in this case methane in primary reformer (steam reformer) tubes, at an elevated temperature, to form a mixture of hydrogen and oxides of carbon.

$$CH_4 + H_2O \leftrightarrow CO + 3H_2$$
$$CO + H_2O \leftrightarrow CO2 + H2 \longrightarrow \text{Reaction at the primary reformer (endothermic)} \quad (1)$$

The first reaction is called the *reforming reaction.* The reforming reaction is endothermic, and needs energy input in the form of fuel firing, which is normally naphtha or natural gas.

Air is then mixed in with the gas stream to give hydrogen to nitrogen ratio of 3:1 at the secondary reformer to produce ammonia. Refer to **APPENDIX A** for the full synthesis process.

$$N_2 + 3H_2 \leftrightarrow NH_3 \quad \text{(Final Product)} \quad (2)$$

2.3 Tube Failures

Failure mechanism of tubes is affected by overheating that might cause creep damage. Amongst factors lead to overheating are investigated in a case study of reformer tubes from a fertilizer plant made of modified HK 40 steel which failed after 4 years. The failure mechanisms and life evaluation are investigated and analysis found that longitudinal cracks found in the tubes were caused by overheating because of inadequate feed flow caused by choking of

damage catalyst. This case study has also proven that overheating during service is primarily responsible for significant degradation in mechanical properties and microstructures of the reformer tubes thus supporting the effort of modeling the primary reformer TMT [6].

Figure 6 :100 % Tube Failure (left) and Tube Rupture (right)

Reformer tubes have had several premature failures in PETRONAS plants that cost millions of ringgit. The examples are [5]:

- PASB had 3 failures until the year 2008
  (RM 14.4 million replacement cost for all 288 tubes, RM 20 million per catalyst batch, and RM 30 million production loss)
- PETRONAS Methanol Labuan (PML) had 100% tube failure in 2000

2.4 Reformer Modeling Techniques

Model is a description of a system to explain the behavior of a system and predicting its responses when inputs are applied. Most of the reformer modeling or simulation involves only mathematical/fundamental modeling which is very complex and might not be feasible to be applied in industry. In this project, empirical model or black box model is to be proposed to predict primary reformer TMT with better accuracy. By the time this predictive model is being developed, there was none empirical model developed to predict the tube metal temperature in a steam reforming process [4].

2.4.1 Fundamental/Mathematical Model

Mathematical/fundamental model uses mathematical equations based on fundamental laws or theories. If a model is perfectly known with its prior knowledge and physical insight, it will be possible to construct a model and thus called white-box model.

Problems with mathematical modeling are, it requires a large time investment before actually applying the techniques, and even some techniques fail when implemented in complex systems, due to the presence of saturation, unknown disturbances, etc. [7]

2.4.1.1 Modeling, Simulation, and Sensitivity Analysis of Steam Methane Reformers [8]

In this journal, mathematical model to calculate temperature, conversion and pressure profiles for static operations in steam-methane reformers was simulated. A rigorous kinetic model describing steam-methane reactions was compared to a first order and an empirical heat distribution model was fitted to describe heat absorbed along the reactor length.

The kinetic models were tested with data from industrial steam-gas reformers. Simulation results agreed with actual plant data for conversion, temperature and pressure. The rigorous model could confidently be used for design analysis, control, and economic evaluation purposes.

This journal helps to understand the fundamental of operation in steam-methane reformers.

2.4.2 Artificial Neural Network (ANN)

Neural network is composed of simple elements operating in parallel and inspired by biological nervous systems. The neural network can be trained to perform particular function by adjusting the values of the connections (weights and biases) between elements (Fig. 7) and is considered as black box model [9].

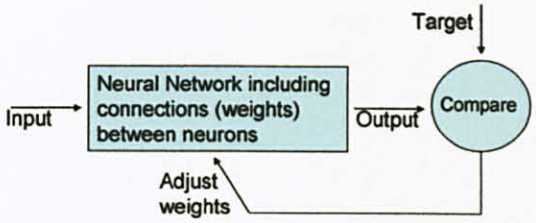


Figure 7 : Neural Network Model as Function Approximation Tool

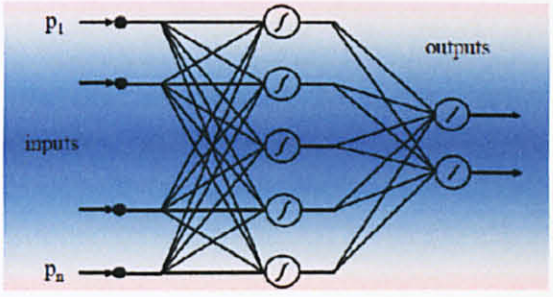


Figure 8 :MLP Structure (2 Layers)

Neural network can be trained to perform complex functions in various fields of application. There are quite a number of network architectures and the learning techniques that have been developed up to this date and being applied in various fields. Several network architectures that available in ANN are feedforward/Multilayer Perceptron (MLP) network, recurrent network, radial basis function (RBF) network and adaptive linear (Adeline) network. In this project, the modeling using the feedforward/MLP backpropagation network, recurrent (Elman) network, and radial basis function (RBF) network are developed. The Elman network is a type of recurrent network that has feedback loops from the output in the hidden layer to the input layer. While RBF network consists of 2 layers and contains the radial basis function in the hidden layer [9].

The model development of ANN Model is performed using MATLAB's Neural Network Toolbox. As the model performance is affected by some parameters such as neural network structure and quality of data pre-processing, these parameters need to be explored in order to obtain an optimal ANN model structure [10]. Few parameters that can be tuned/ varied to minimise the error goal are:

- Network architecture (feedforward/MLP, recurrent, RBF)
- Activation/transfer function (logsig, tansig, purelin)
- Performance indicator/function (mse, msereg, sse, mae)
- Training styles (incremental, batch)
- Training algorithms (backpropagation; traincgp,trainlm)

This project describes the approaches taken by mean of data mining to observe the effect of changing those parameters on the model performance.

2.4.2.1 Neural Networks Cartridges for Data Mining on Time Series [10]

Neural Network is the technique used for time series analysis. The performance of neural networks is affected by some parameters such as neural network structure and the quality of the data preprocessing. However, the manual establishment of different neural networks configurations for selecting the ones may be time consuming and prone to trigger errors.

This conference paper proposes the creation of neural networks cartridges by means of data mining activities to obtain an optimal neural network structure. The idea presented in this paper is used to perform neural network modeling.

2.4.2.2 Neural Network Predictive Control (Application)

One of the reasons why the development of the predictive model is needed in predicting the primary reformer TMT is due to the expensive cost of installing physical sensor. It is also to avoid nagging issues such as maintenance. A predictive model (soft sensor) is based on the use of software technique to

determine the value of a process variable in contrast to a physical sensor (e.g. Resistance Temperature Detectors) which directly measures the value of process variable. Neural network-based soft sensor can be embedded into existing processing control system without significant cost incurred in traditional implementation (installing physical sensors at all tubes can be bulky).

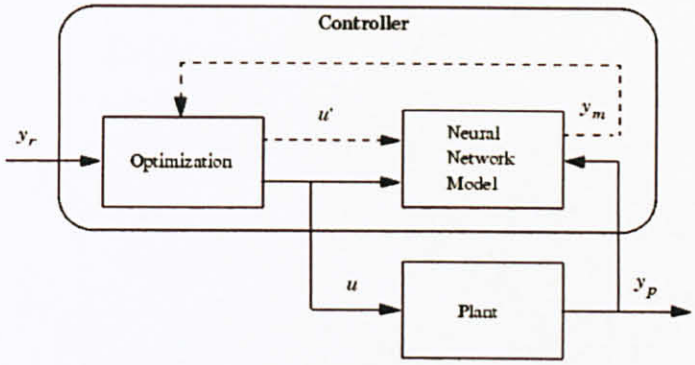


Figure 9 : Model Predictive Control

In implementing control strategy, the neural network model is used to predict future plant performance. The controller then calculates the control input that will optimise plant performance over a specified future time horizon. An optimization block also used to optimise and to adapt the neural network model accordingly [9].

In this project, the neural network model will only be trained offline (batch) by using real-time data obtained from PASB earlier. The developed model can be embedded into existing control system as part of the control strategy which is the later step/future work.

2.4.3 System Identification (AutoRegressive Exogenous)

System identification is the task of inferring a mathematical description, a model based on the observed input-output data without prior knowledge of the process side (black box model). The parameters of a given model are adjusted for the predicted outputs to approximate the measured ones. There are variations of

models in system identification such as parametric model structure and plant model [11]. In this project, a parametric model structure represented by differential equation is being used.

AutoRegressive eXogenous (ARX) model is selected in the model development work. System identification using ARX model, for Multi input Single Output (MISO) is developed using MATLAB's System Identification Toolbox. ARX is a model that relates the current output, *y* to a finite number of past outputs and inputs, *u* [11].

$$y(t) + a_1 y(t-1) + .. + a_{na} y(t-na) = b_1 u(t-nk) + .. + b_{nb} u(t-nk-nb+1) \qquad (3)$$

The structure is defined by three integers, *na*,*nb*, and *nk* (orders) which is equal to the number of poles and *nb-1* is the number of zeros, while *nk* is the pure time delay in the system. The coefficients *a* and *b* in the ARX model structure are estimated using Least Square Method [11].

2.4.3.1 Black box modeling of Steam Temperature (ARX Model) [12]

This paper describes the black box modeling of steam temperature based on the real-time data. The AutoRegressive eXogenous (ARX) model structure was selected in this initial work. The real time data were obtained using computer-based data acquisition system from a pilot scale distillation column. This research covers the area of understanding input-output behavior of a process, predicting future response, developing control system and tuning algorithms and filtering signal. The System Identification Toolbox in MATLAB package was used for model development and validation.

This conference paper provides general understanding of system identification techniques that will be implemented in modeling the primary reformer TMT.

2.4.4 Multiple Linear Regression (MLR)

Multiple Linear Regression (MLR) is one of the most used methods for forecasting/predicting and is widely used to fit the observed data and to create models in many research fields such as in psychology (Ansiau et al., 2005) [13]. MLR attempts to build the quantitative relationship between a group of predictor variables (columns of x) and a response, y (which also categorised as black box model). The input-output relationship can be given by:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots \beta_n x_n + \varepsilon \quad (4)$$

The objective is to find suitable coefficient values, β_n using Least Square Method in order to build an adequate model that can predict TMT. Apart from predicting future values, the model built is also useful for:

- Knowing the direction of the effect (increasing x will increase or decrease y)
- Understanding which predictors (x) have the greatest effect

2.5 Performance Index

In assessing the performance model, the best fit criterion (FIT), which indicates a better model for a higher number of fit values has been used and is calculated by software [4].

$$FIT = \left(1 - \frac{\sum_{i=1}^{n} |y_i - \hat{y}|}{\sum_{i=1}^{n} |y_i - \bar{y}|}\right) \times 100\% \quad (5)$$

$\hat{y}_i$ is the predicted output and $\overline{y}_i$ is the average measured output.

Root Mean Square Error (RMSE) is also being used in comparing models' performances. It is based on the mean square distance between the measured output, $y_i$and predicted output $\hat{y}_i$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2} \quad (6)$$

In this modeling, it is assumed than the performance of validation set is much more important that the training/estimation set to show how the unknown (fresh) set of data are to be generalised in yielding output.

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

Below depicts the general flowchart used for modeling primary reformer TMT in offline mode (batch) using different black box model approaches.

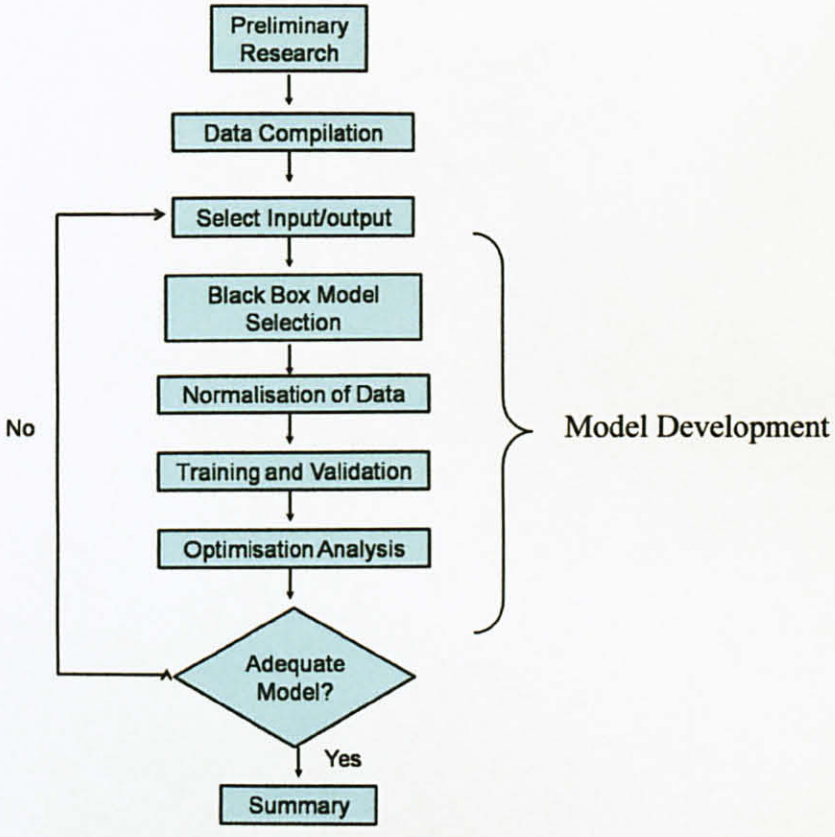


Figure 10 :Project Flow Chart

Refer to **APPENDIX B** for Gantt chart to see the timeline of this project.

3.2 Selection of Process Variables

Selection of process variables is conducted by performing statistical analysis. In this project, the critical process variables have been selected based on the Pearson product-moment correlation coefficient where the mathematical expression is shown below:

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_x \sigma_y} \quad (7)$$

Where x is the input, y is the output, and σ is the standard deviation. The correlation coefficient is basically the ratio of covariance and product of standard deviations of two variables. The correlation coefficient is calculated in MATLAB by using '**corrcoef(*x*,*y*)**' function. It measures the statistical relationship between two variables in the range of $-1 \leq \rho \leq +1$. For perfect positive correlation (+1), the value will be positive one. For the perfect inverse relation, the correlation coefficient indicates negative one (-1). If there is no relationship between two variables, the correlation coefficient value will indicate zero value [4].

3.3 Artificial Neural Network (ANN) Model Development

First Phase

In the development of the neural network model, procedures below have been performed to study their effects on the model performances:

- Establish default neural network architecture
- Using different training algorithms to test which performs the best for the application
- Using different normalisation method and performance function
- Changing the number of neurons in the hidden layer
- Outlier Removal

The data division for this development is 50:50 for the estimation/training and validation sets.

Second Phase

The second phase of the model development of the neural network will be experimental. Thousands of iterations are performed to see the results of changing certain parameters that affect the performance of the neural network. Three different architectures are involved in this second phase development:

a. MLP network (backpropagation algorithm)
b. Recurrent network (Elman)
c. Radial basis function network (RBF)

The results will be shown in the results and discussion section.

3.4 System Identification (ARX model) Model Development

The modeling of TMT using parametric model AutoRegressive Exogenous (ARX) is done via MATLAB's System Identification Toolbox. Below outlines the approaches taken in developing ARX model:

- Pre-processing of data. This involves filtering and removing trends and outliers
- Selecting range. The data set is divided into estimation/training and validation data set by 50:50
- Parameters estimation. The orders of the ARX model are defined and the parameters can be obtained using Least Square Method that find a set of coefficient that minimises the square error to all known dataset
- Model validation. This is done using a part data (validation set) that has yet being used
- Examine the model performance. Assessing the performance of the model, if it is not good enough, the process is repeated starting from the parameters estimation

The results based on ARX modeling will be presented in the results section.

3.5 Multiple Linear Regression (MLR) Model Development

The modeling of MLR is done using algorithms in MATLAB's Statistical Toolbox. The data set is also divided 50:50 for estimation/training and validation sets.

3.6 Tools and Equipment

The tools that are being used in developing the black box model to predict TMT in this project are:

a) MATLAB Toolbox:

i) Neural Network Toolbox

ii) System Identification Toolbox

iii) Statistical Toolbox

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Data Compilation and Level Selection [4]

Two sets of input-output data have been provided from PASB. For the inputs, the readings of the process variables are taken from the distributed control system (DCS). As for the outputs, the temperature readings of the tube metal temperature are taken manually using pyrometer via peepholes at level 1 and level 3 (Fig. 11). Refer to APPENDIX H for the data manual entry form [4].

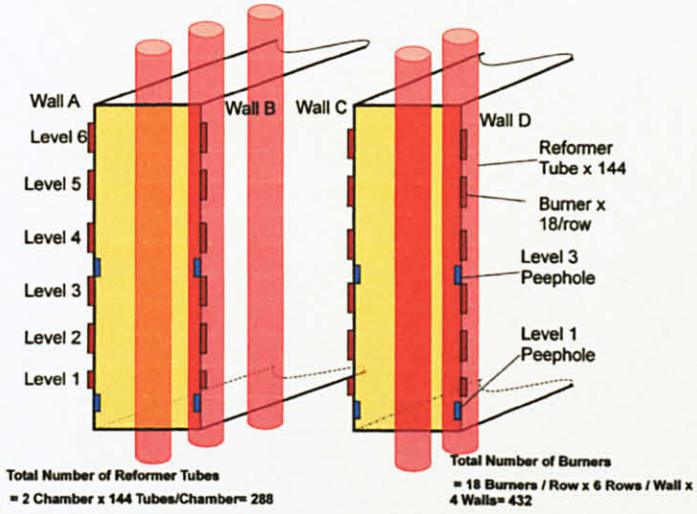


Figure 11 : Levels and Peepholes' Locations

The details of the data descriptions are shown as below:

Table 1 :Description of PASB Data

Set	No. of Process Variable	No. of Data		Process Variable Sampling Time	Period
		Level 1	Level 3		
A	19	-	187	1 hour	June 07-June 08
B	19	27	43	1 hour	July 08-October 08

Both set of data are compiled and pre-processed to be used for model development. Followings are some of the assumptions made:

a) Process variables readings are taken between 11pm to 2am (from DCS) synchronously with the manual measurement of TMT (pyrometer)
b) The process variables and TMT values used in the data set are the average values (average temperature of 144 tubes - for each chamber)
c) Unique model will be developed for each chamber (chamber 1 and chamber 2)
d) All the average values of process variables and TMT are paired, compiled, and the irrelevant data (outliers, zero readings, spike) are removed
e) The final data set consist of 200 input-output pair and will be used later in model development
f) The data are covering the whole operating region

Because the data are taken for both level 1 and level 3, the higher average TMT values are selected to be used as outputs. Below graph shows the average TMT distribution for different level and chamber (up until 43 data sets only):

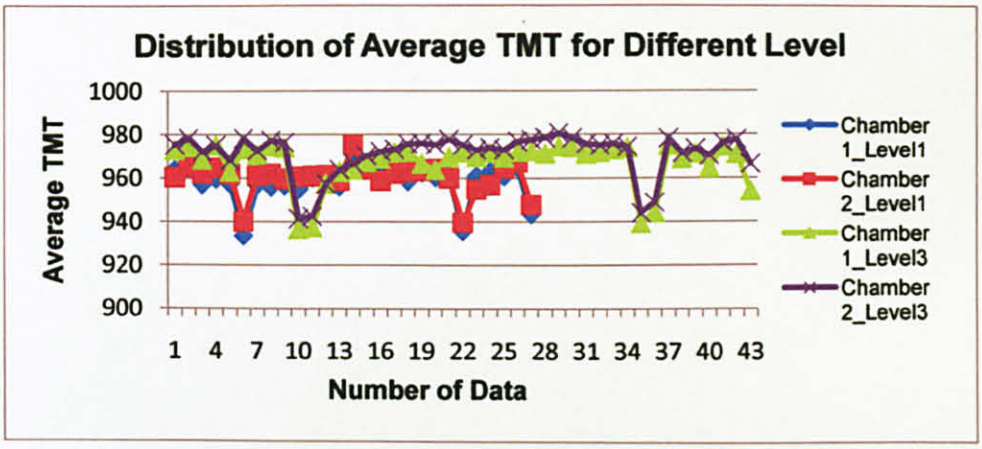


Figure 12 :Average TMT Distribution for Level 1 and 3

Based on the observation, average TMT values on the level 3 have higher temperature readings compared to level 1. Thus, data (TMT) from level 3 will be used as outputs for the model (chamber 1 and chamber 2) to be developed.

4.2 Inputs/Process Variables Selection

As explained in methodology section, the selection of process variables is based on the correlation coefficient that is the relationship between two variables.

From the overall 19 process variables, the usage of the process variables as inputs in modeling is reduced to only 7 process variables. Selections are based on the highest correlation coefficients between each input and each output respectively (**APPENDIX C**). The other minimum criterion in selecting critical process variables is the absolute correlation coefficients must not be less than 0.6. The inputs selected and correlation coefficients are displayed in the Table 2 and are depicted in the Fig. 13 below.

Table 2 :Process Variables and Correlation Coefficient

				Correlation Coefficient	
Num.	**Tag Number**	**Process Variable**	**Unit**	**Chamber 1**	**Chamber 2**
1	FIA-1216	Combustion air flow	Nm^3/h	0.784	0.779
2	FRCA-1202	Natural gas feed flow	Nm^3/h	0.795	0.803
3	FICA-1251	Total calorific energy	GJ/h	0.766	0.767
4	TRA-1232	Reformer outlet temperature	°C	0.799	0.773
5	PICA-1253	Fuel pressure	kg G	0.679	0.626
6	FFRA-1208	S/C ratio	-	0.791	-0.796
7	TIA-1212	Feed temperature	°C	0.704	0.711

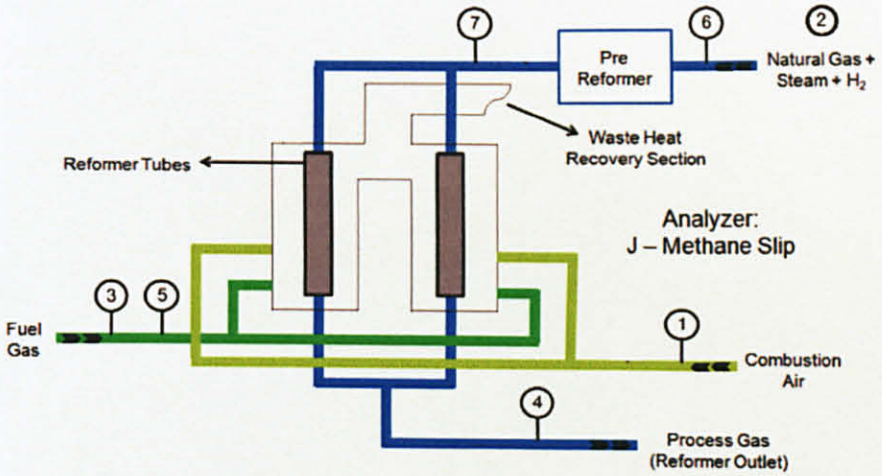


Figure 13 : Process Flow and Instruments' Locations [4]

4.3 First Phase Artificial Neural Network (ANN) Modeling

ANN model is developed by using MATLAB's Neural Network Toolbox (Refer to **APPENDIX D** for programming code). Performance of an artificial neural network (ANN) model is depending on some parameters; the structure and the preprocessing of the data. In ANN model development, several approaches are performed heuristically to obtain the optimal structure of a neural network model and are aforementioned in methodology section

i. Establish default neural network structure

Early works done by V. N. Patel, 2009 [4] has developed a tuned structure of ANN model with the lowest RMSEs for the validation set; 5.463 for chamber 1 and 5.233 for chamber 2 respectively. The default structure is represented as Table 3 and will be further tuned to get much more promising results

Table 3 : Default ANN Architecture

Parameters	Description
Network Type	Feed forward/MLP
Hidden Layer	1
Number of Neurons	18 (hidden layer)
Hidden Layer Transfer Function	Logsig
Output Layer Transfer Function	Purelin
Learning Rate	0.2
Training Function	ALL
Normalisation	0 to 1
Performance function	MSE
Final Performance Indicator	RMSE
Data Structure	200 data sets (divided 50:50)

ii. Using different training algorithms

By using different variation of training/learning algorithms listed in Table 4 [9] the ANN model is trained and simulated to see the pattern that could emerge in optimising the ANN model's performance. The development process is done

only for chamber 1's TMT and the performance is only assessed on the validation set, assuming the validation performance results are more important than the training performance results (Table 5).

Table 4 : Variation of Training Algorithms/Training Functions

Training Function	Algorithm
'trainrp'	Resilient Backpropagation
'traingda'	Variable Learning Rate Backpropagation
'traingdx'	Variable Learning Rate Backpropagation with momentum
'traincgf'	Fletcher-Powell Conjugate Gradient
'traincgp'	Polak-Ribiére Conjugate Gradient
'traincgb'	Conjugate Gradient with Powell/Beale Restarts
'trainscg'	Scaled Conjugate Gradient
'trainbfg'	BFGS Quasi-Newton
'trainlm'	Levenberg-Marquardt

Table 5 :ANN Performances on Algorithms (Chamber 1)

Training Function	RMSE (Train)	RMSE (Validate)
'trainrp'	5.6446	7.6812
'traingda'	6.6200	5.1977
'traingdx'	6.1337	6.0164
'traincgf'	6.0002	6.9162
'traincgp'	5.8960	7.0486
'traincgb'	5.9615	6.9962
'trainscg'	5.2829	7.4656
'trainbfg'	5.6364	7.7416
'trainlm'	4.2452	11.9973

Based on the validation set for the TMT chamber 1's results, Variable Learning Rate Backpropagation (traingda) is observed to perform the best with the lowest validate RMSE, while taking maximum epoch (1500) before the training stopped. The result for the Levenberg-Marquardt (trainlm) shows the worst performance with the highest validate RMSE. Levenberg-Marquardt in this case,

did not give promising results (initially) even though in many cases (research), Levenberg-Marquardt is able to obtain lower root mean square error values compared to the other algorithms. The reason is still unclear, but this might be due to the complex mathematical calculation. However, these initial results draw no significant conclusion as the development process is still in the early phase. The maximum epoch is increased to 2000 to give ample time for the result to converge on the next steps.

iii. Using different normalisation method and performance function

Normalisation of data helps to prevent attributes with large ranges from outweighing attributes with smaller ranges as well as speeding up the computation time [10]. The effects of changing the normalisation method using 0 to 1, 0.1 to 0.9 and 0.2 to 0.8 ranges are studied in a journal that indicates the usage of those types of normalisation method outperform the performance of the conventional -1 to 1 normalisation in terms of getting lower RMSE values [14]. In this project, the data are normalised into 0.1 to 0.9 range to see the effect of performance. The performance function is also changed from the previous 'MSE' to 'MSEREG' as to perform regularisation technique intended to produce smaller weight and biases thus forcing the network response to be smoother [9]. The results are tabulated in the Table 6 below:

Table 6 :ANN Performances on Performance Function

Training Function	RMSE (Train)	RMSE (Validate)
'trainrp'	6.4922	5.3060
'traingda'	6.7894	5.1145
'traingdx'	6.4765	5.3237
'traincgf'	6.4065	5.4861
'traincgp'	6.3903	5.4618
'traincgb'	6.3911	5.4600
'trainscg'	6.3898	5.4620
'trainbfg'	6.3898	5.4620
'trainlm'	6.1174	6.4963

Comparing with the previous results, the degradations of performances can be seen for the training's models while overall improvements on the validation's models. As this ANN modeling is done by mean of data mining, the optimisation process is continued without drawing any conclusion from these results.

iv. Changing the number of neurons in the hidden layer

The iteration of changing the number of neurons from 1 to 100 neurons for the hidden layer is performed and analysed on every training algorithm stated above. The results show that for every training algorithm excluding the Variable Learning Rate Backpropagation (traingda and traingdx), and Levenberg-Marquardt (trainlm) has the same trend whereby as the number of neurons increase, the RMSE decreases until it reaches an optimum number of neurons, and then RMSE starts to increase again (Fig. 14). The optimum number recorded is 4 neurons. The performances of the training algorithms with the optimum number of neurons in terms of RMSE of the validation set are displayed in Table 7.

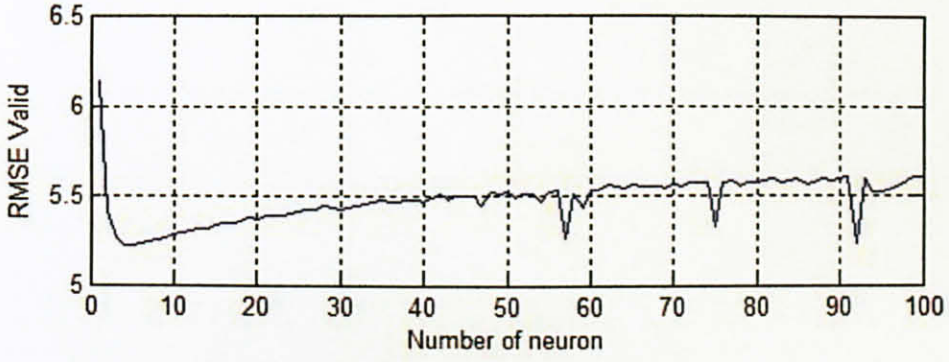


Figure 14 : RMSE with Increasing Neuron Numbers in Hidden Layer

Table 7 :ANN Optimal Performances With Changing Number of Neurons in Hidden Layer

Training Function	RMSE (Train)	RMSE (Validate)
trainrp	4	5.1408
traingda	21	5.0731
traingdx	49	5.1154
traincgf	4	5.1409

traincgp	4	5.1404
traincgb	4	5.1409
trainscg	4	5.1408
trainbfg	4	5.1408
trainlm	2	5.1184

The results for the Variable Learning Rate Backpropagation (traingda and traingdx), and Levenberg-Marquardt (trainlm) show oscillation of RMSE values when the number of neurons is increased. The results for these three algorithms are then omitted first as they are not synchronised with the other training algorithms.

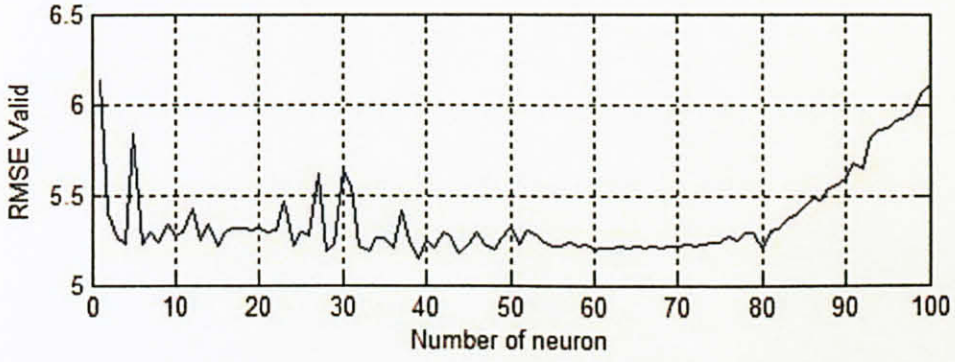


Figure 15 : Oscillation of RMSE (Validate) for Traingdx

From the results, Polak-Ribiére Conjugate Gradient (traincgp) performs the best with the validate RMSE value is 5.1404 and will be used for further optimisation.

v. Outlier Removal

The ANN model is being re-modeled and trained using Polak-Ribiére Conjugate Gradient (traincgp) and the validation error's plot is observed as Fig. 16 below. From the observation, at the validation set, there is an outlier at index 149 of the total 200 data. The input-output data pair is then removed leaving altogether 199 pair of data. Table 8 describes the latest ANN structure. The effect of the removal of the outlier is again tested and the summary of the performance results are shown in Table 9.

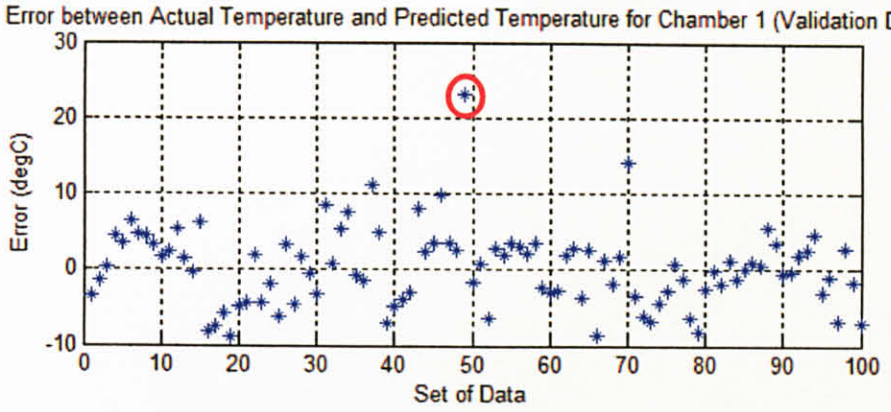


Figure 16 : Validation Error's Model Scatter Plot

Table 8 : The Latest Neural Network Architecture

Parameters	Description
Network Type	Feed forward/MLP
Hidden Layer	1
Number of Neurons	4 (hidden layer)
Hidden Layer Transfer Function	Logsig
Output Layer Transfer Function	Purelin
Learning Rate	0.2
Training Function	TrainCGP
Normalisation	0.1 to 0.9
Performance function	MSEREG
Final Performance Indicator	RMSE
Data Structure	199 data sets (outlier removed)

The structure above yield results as tabulated in the below.

Table 9 :ANN Model Performance

Index	Chamber 1		Chamber 2	
	Train set	Validate Set	Train set	Validate Set
RMSE	6.9064	4.6156	7.4993	4.3553
FIT (%)	45.5043	47.9196	42.4455	49.4076

In evaluating ANN performance, the validation set performance is considered much more important than the training set performance. The results improved significantly in terms of RMSE performance as the outlier in the dataset is removed which might show generalisation error. The optimum results up until this method is executed are RMSE for chamber 1 is 4.6165 and chamber 2 is 4.3553 respectively for the validation set and 15-16 % improved in accuracy can be observed from the start of development of the ANN Model. FIT performances show quiet good results as they are higher than 40% which indicate good reproducibility. The plot for the ANN model for the estimation/training and validation set of the chamber 1 are shown as Fig. 17:

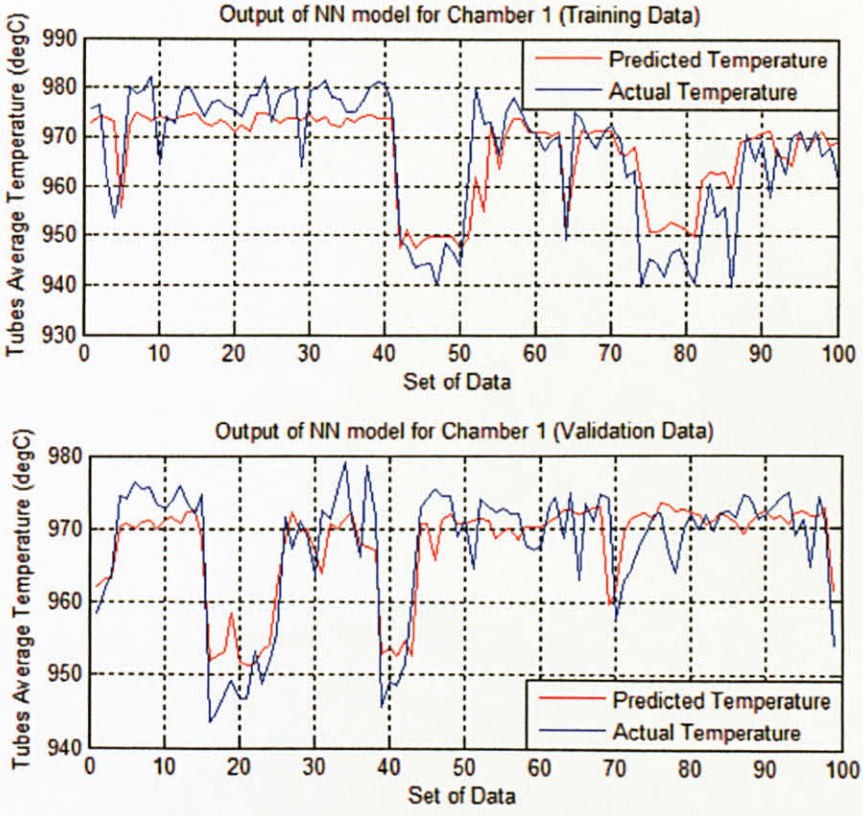


Figure 17 :ANN Model Plot (Training and Validation) for Chamber 1's TMT

4.4 Second Phase Artificial Neural Network (ANN) Modeling

In this second phase ANN development, different ANN architectures (apart from MLP) are developed. The ANN model (MLP and Elman) is **repeatedly trained and simulated** with the increasing number of neurons for different training algorithms. The lowest value of validate RMSE values are recorded with the corresponding number of neurons for each training algorithm.

The feedforward/MLP network, recurrent network (Elman), and RBF network models are generated by means of data mining and trial and error. This involves thousands of iterations that will serve to see the effects of changing the structures on performance. The data used in this iteration are the previous 199 data (divided 50:50), with the normalisaton 0.1 to 0.9.

MLP Model (Second Phase)

The number of neurons is changed (1 until 100 neurons) for each algorithm to see which gives **the best RMSE and FIT values.** For the MLP network:

Table 10 : Chamber 1's MLP Network Iterations (1 to 100 Neurons)

Algorithm	RMSE (Validate)	FIT % (Validate)	Num. of Neurons
trainrp	4.5884	48.2272	6
traingda	4.4667	49.6003	21
traingdx	4.4614	49.6598	53
traincgf	4.5857	48.2570	6
traincgp	4.5816	48.3032	7
traincgb	**4.4569**	49.7102	92
trainscg	4.5883	48.2275	6
trainbfg	4.5883	48.2275	6
trainlm	4.6284	47.7751	2

Table 11 Chamber 2's MLP Network Iterations (1 to 100 Neurons)

Algorithm	RMSE (Validate)	FIT % (Validate)	Num. of Neurons
trainrp	4.2944	50.1149	8
traingda	4.1961	51.2570	7
traingdx	4.1744	51.5086	53
traincgf	4.0824	52.5778	81
traincgp	4.2232	50.9427	95
traincgb	**4.0801**	52.6045	99
trainscg	4.2981	50.0720	7
trainbfg	4.2982	50.0709	7
trainlm	4.3268	49.7388	2

The numbers of neurons that produce the lowest RMSEs for the validation set are not the same for all algorithms. However, the **overall improvement** on the lowest validate RMSE values (Table 10 and Table 11) can be seen after the removal of the outlier leaving the data to be only 199 (ANN model – phase 1). The best algorithm to train the model in predicting TMT can be any of the algorithms, but it depends on the application, speed and the amount of the memory that will be used [9].

The algorithm that produces slightly the best results in term of the lowest RMSE for the validation set is **traincgb**. However, the number of neurons in the hidden layer needed (in the range of 1 to 100 neurons) is not consistent (not the same) for both chamber 1 and chamber 2's models which are 92 neurons for chamber 1 and 99 neurons for chamber2's models.

Thus, the traingdx models (chamber 1 and chamber 2) are taken to represent MLP network because they have **consistent number of neurons** to produce the lowest RMSEs results (using traingdx algorithm only). The traingdx models with 53 neurons in the hidden layer are re-modeled (refer **APPENDIX D** for weight and bias values) and yields results as in Table 12:

Table 12 : Second Phase ANN (MLP) Models' Performance

Index	Chamber 1		Chamber 2	
	Train set	Validate Set	Train set	Validate Set
RMSE	6.7720	4.4614	7.3060	4.1744
FIT (%)	46.5651	49.6598	43.9289	51.5086

Elman Model

The same steps above are repeated to model the recurrent network (Elman). The MATLAB's code used is still the same (**APPENDIX D**), and the differences are the function to create the Elman network which is 'newelm' and the maximum number of neurons that is iterated **are only up to 35 neurons** in order to shorten the iteration time. The iteration results are shown as table below:

Table 13 : Chamber 1's Elman Network Iterations (1 to 35 Neurons)

Algorithm	RMSE (Validate)	FIT % (Validate)	Num. of Neurons
trainrp	4.5852	48.2629	4
traingda	4.4375	49.9294	11
traingdx	4.4722	49.5379	34
traincgf	4.5852	48.2624	4
traincgp	4.5853	48.2620	4
traincgb	4.5853	48.2618	4
trainscg	4.5852	48.2628	4
trainbfg	4.5852	48.2628	4
trainlm	4.6228	47.8382	2

Table 14 Chamber 2's Elman Network Iterations (1 to 35 Neurons)

Algorithm	RMSE (Validate)	FIT % (Validate)	Num. of Neurons
trainrp	4.292176	50.1410	4
traingda	4.229651	50.8673	20
traingdx	4.221277	50.9646	47
traincgf	4.29153	50.1485	4
traincgp	4.288093	50.1884	4
traincgb	4.293364	50.1272	4

trainscg	4.292178	50.1410	4
trainbfg	4.292164	50.1412	4
trainlm	4.321999	49.7946	2

For the algorithms excluding traingda, traingdx and trainlm, the result for the lowest RMSEs that can be obtained are consistent which is by using 4 neurons. However, the predicted TMT values for the Elman are inconsistent for different iterations. For example even if two Elman networks, with the same weights and biases, and are given identical inputs at a given time step, their outputs can be different due to different feedback states [6]. This will incur problem in predicting TMT in the future as the values predicted might be deviated from its initial pattern. The full results using trainscg function is shown in **overall performance results section (section 4.7)**

RBF Second Phase Model

Radial basis network is one of the major architecture of ANN. It can be designed with the function 'newrbe' and no number of neurons **needs to be defined**. This function can produce a network with zero error on training vectors. It is called in the following way:

net = newrbe(P,T,SPREAD) %P is input, T is target and, Spread is constant

Kindly see **APPENDIX E** for the MATLAB's code developed. The parameters that need to be tuned in this RBF model is the **spread constant** that will influence the bias of the first layer [6]. By tuning the spread constant in modeling the chamber's 2 TMT first, the results are as below:

Table 15 : Chamber 2's Performance on different SPREAD constant

SPREAD constant	RMSE (Train)	RMSE (Validate)
0.180	1.0000	8.5839
0.185	1.0000	**8.1673**
0.190	1.0000	8.3474
0.200	1.0000	10.6662

From the results above, the performance for the chamber 2 model is at the lowest when the spread value equals to 0.185. Thus the spread constant value is used again to model for the chamber 1's TMT.

Table 16 : Models' Performances with 0.185 SPREAD constant

Index	Chamber 1		Chamber 2	
	Train set	Validate Set	Train set	Validate Set
RMSE	1.0000	52.6360	1.0000	8.1673
FIT (%)	100.00	-493.9181	100.00	5.1261

Chamber 2's TMT model by using spread constant 0.185 has the lowest RMSE for the validation set value with 8.1673, but the FIT is just only 5% representing very poor reproducibility and could not generalised well. For chamber 1, by using the same constant, the result is even worse, with the high value of validate RMSE 52.6360, and with the negative value of FIT, and from graph below show the inverse response of the predicted values of the Chamber 1's validation TMT model:

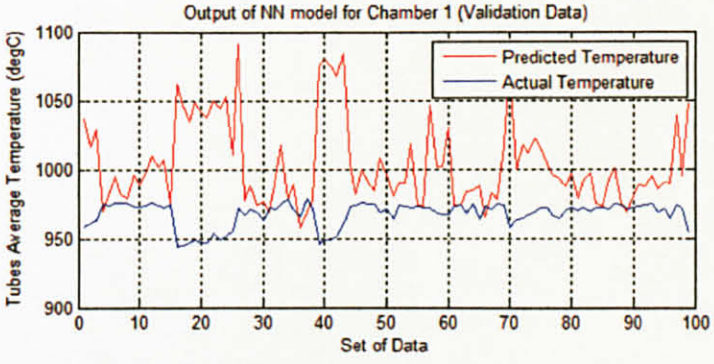


Figure 18 : Inverse Response of the Chamber 1's TMT Model

The RBF model failed to predict primary reformer TMT. A way to improvise the RBF structure shall be explored to developed a better model that can predict TMT.

From the modeling of three architecture above (MLP, ELM and RBF), the best results for respective architecture are summarised as below:

4.5 System Identification Modeling (AutoRegressive Exogenous)

The ARX model is developed using the Graphical User Interface in the MATLAB's System Identification Toolbox that is friendly user (Fig.19). Two models with different orders have been developed.

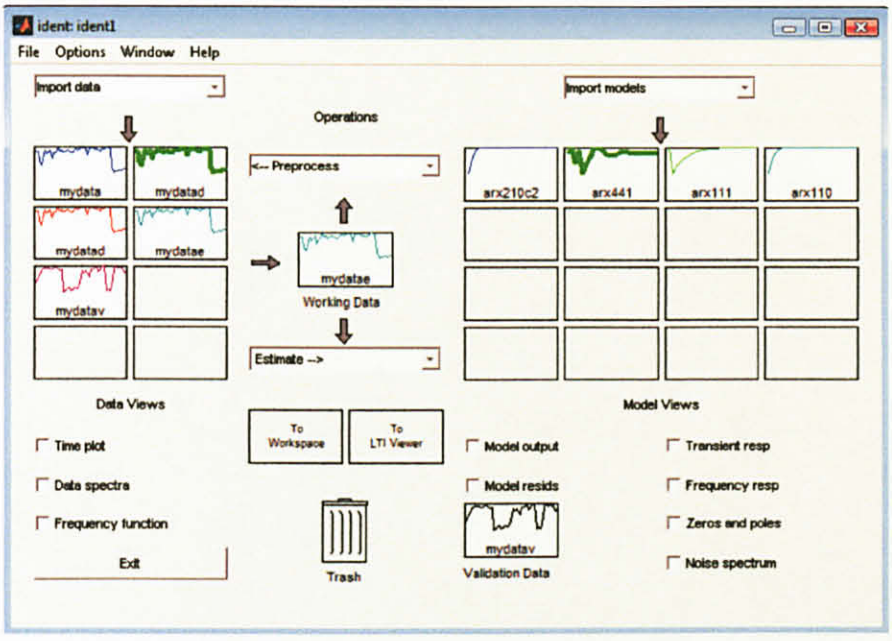


Figure 19 : System Identification Toolbox (GUI)

Referring to the equation (1), the orders of the first ARX model have been defined as *na*=2, *nb*=1, *nk*=0, and the second ARX model as *na*=4, *nb*=1, *nk*=0, respectively for its zeroes, poles and delays. The parameters are then estimated and **imported** into MATLAB's workspace to be re-modeled using the **199** pair of data (100 for training and 99 for validation). Refer to **APPENDIX F** for the programming code and mathematical coefficients. For the sake of the simplicity, the ARX's first model is called ARX210 and the second model as ARX410. The sampling interval is set to 1.

The performance indexes used for ARX models are also FIT and RMSE. RMSE and FIT are being used as to measure the performance of the model, to compare with the previous ANN model and MLR model later. The performances of the ARX models are tabulated in Table 17 and Table 18:

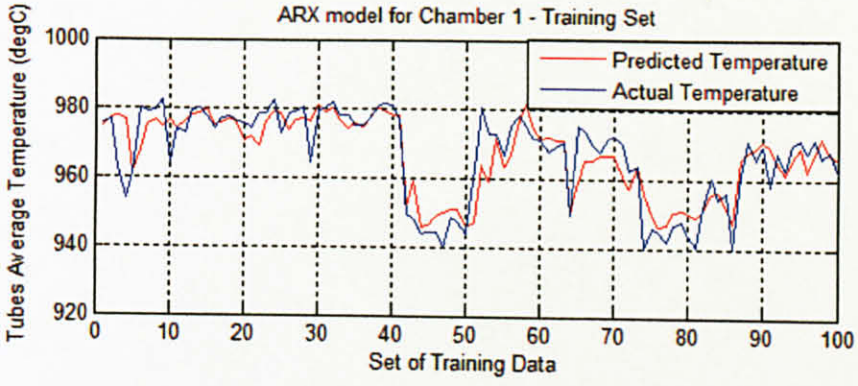
Table 17 : ARX210 Models' Performances

Index	Chamber 1		Chamber 2	
	Train set	Validate Set	Train set	Validate Set
RMSE	6.3592	7.2317	7.1070	7.0865
FIT (%)	49.8225	18.4325	45.4560	17.8650

Table 18 : ARX410 Models' Performances

Index	Chamber 1		Chamber 2	
	Train set	Validate Set	Train set	Validate Set
RMSE	6.2890	6.2425	7.0549	5.5540
FIT (%)	50.3761	29.5828	45.8563	35.6270

In doing comparison, it is assumed that the performance of the validation set is much more important than the estimation/training set. As can be seen in Table 17 and Table 18 above, it can be seen that ARX410 model outperforms ARX210 model for both chamber 1 and chamber 2 predictions in the validation set. The RMSE for ARX410 for chamber 1 and chamber 2 is much more lower that ARX210. Whereas the best fit criterion (FIT) for ARX410 in validation sets produced much more higher results than ARX210. The ARX410 is thus taken as benchmark representing system identification, ARX model. The prediction plot for the estimation/training and the validation data for chamber 1 only are shown in Fig. 20 and Fig. 21:



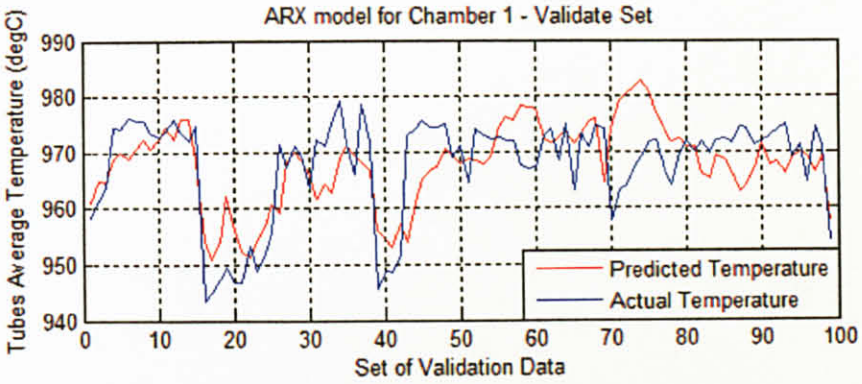


Figure 20 : ARX210 Model Plot (Training and Validation) for Chamber 1's TMT

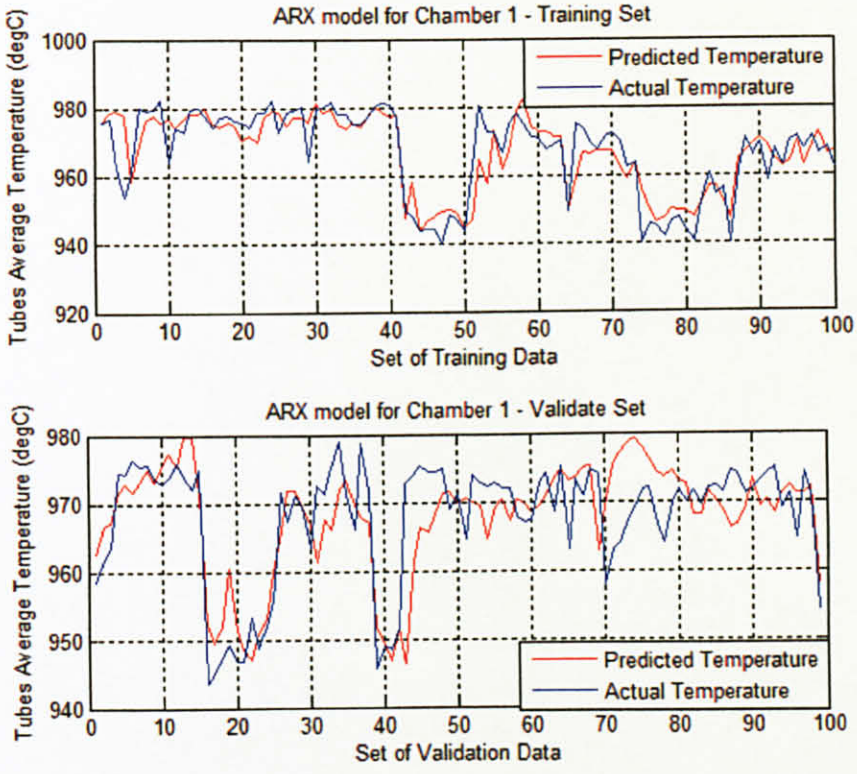


Figure 21 : ARX410 Model (Training and Validation) Plot for Chamber 1's TMT

4.6 MLR Modeling

For the MLR modeling, it is rather straight forward. The computational works done by MATLAB determine the coefficients, β of the MLR model. The results are tabulated in Table 19 in terms of RMSE and FIT values and the prediction plots in Fig. 22 below for chamber 1 only. The programming code and coefficient values are shown in **APPENDIX G.**

Table 19 : MLR Models' Performances

Index	Chamber 1		Chamber 2	
	Train set	Validate Set	Train set	Validate Set
RMSE	6.1068	6.8690	6.9229	6.0438
FIT (%)	51.8140	22.5150	46.8689	29.9502

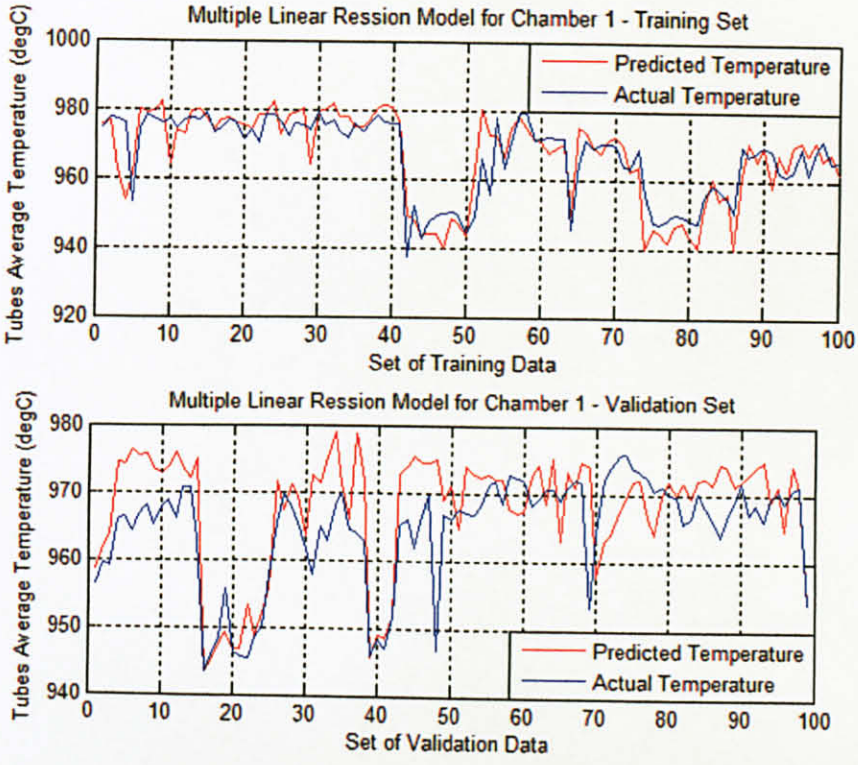


Figure 22 : MLR Model Plot (Training and Validation) for Chamber 1's TMT

The ARX model (ARX 410), with orders *na*=4, *nb*=1, *nk=0* still outperforms the MLR model. This might indicate that the model built by differential equation defines better model than using linear equation.

However, ANN model still outperforms both ARX and MLR models in terms of comparing both RMSE and FIT values of the validation sets.

4.7 Overall performance results

The overall performance results consolidating the first and second phase can be summarised as in the Table 20 below:

Table 20 : Summary of TMT Models' Performances

Model		Chamber	RMSE (Train)	RMSE (Validate)	FIT % (Train)	FIT % (Validate)
ANN	MLP	Chamber 1	6.7720	4.4614	46.5651	49.6598
		Chamber 2	7.3060	4.1744	43.9289	51.5086
	Elman	Chamber 1	6.6699	4.5852	47.3709	48.2627
		Chamber 2	7.3206	4.2922	43.8166	50.1411
	RBF	Chamber 1	1.0000	52.6360	100.00	-493.9181
		Chamber 2	1.0000	8.1673	100.00	5.1261
ARX		Chamber 1	6.2890	6.2425	50.3761	29.5828
		Chamber 2	7.0549	5.5540	45.8563	35.6270
MLR		Chamber 1	6.1068	6.8690	51.8140	22.5150
		Chamber 2	6.9229	6.0438	46.8689	29.9502

From the results above, it was observed that feedforward/MLP architecture has produced the best models with the lowest RMSE (validate) and good reproducibility (FIT). The RMSEs for training set are **6.7720** and **7.3060** respectively for chamber 1 and chamber 2. The RMSEs for the validation set are **4.4614** and **4.1744** respectively for chamber 1 and chamber 2. The radial basis function (RBF) model produces the worst performance and has poor reproducibility.

In comparing with the previous works done (V.N. Patel), the performance of the model in predicting TMT improves significantly from the initial performance where RMSE (validation) are 5.463 and 5.233 for chamber 1 and chamber 2 respectively to 4.4614 and 4.1744 in this project.

The accuracy or performance of the model to predict primary reformer TMT could be further improved from time to time. A robust primary reformer TMT models can be integrated into existing control system to implement relevant control strategy (future work) if the predictive models developed are good enough.

CHAPTER 5
CONCLUSION & RECOMMENDATION

5.1 Conclusions

Primary reformer is critical equipment used in ammonia, methanol and hydrogen plants. The steam reforming process occurs inside the tube is an endothermic process that needs external heat for the reaction to take place. Overheating could lead to reformer tubes' failures, thus a model that can predict the primary reformer tube metal temperature (TMT) shall be developed to provide a sufficient monitoring and control system.

This project describes the development of the model of the primary reformer TMT by using three different empirical models via MATLAB. The models developed are ARX model, MLR model and ANN model (MLP, ELM and RBF). The modeling done for the ARX model was done via trial and error method. For the ANN model, heuristic approaches by mean of data mining are taken to serve as tool to improve the performance of model. This project proved that heuristic approach executed in systematical way to setup an optimal neural network structure is viable. The process variables used as inputs in developing the model are selected based on the correlation coefficient values.

In comparison with the ARX model and MLR model, ANN model is observed to perform the best, in terms of getting the lowest RMSE and the highest FIT for the validation sets of the two chambers. Between MLP, RBF, and Elman network (ANN's architectures), the MLP network can be seen as the most transparent and the best way to develop robust TMT model.

An adequate model is needed to predict the TMT in a reformer to prevent overheating of reformer tubes that could lead plant to unscheduled plant downtime and losses.

The project flow is as planned, and meets its objectives.

5.2 Recommendations

The primary reformer TMT modeling is a continuous development process and there are lots of things could be done to further improve the predictive model. The modeling works that have been conducted so far are focusing more towards improving the performance based on the model structure (ANN particularly). The possibility to apply genetic algorithms to speed up the process to converge at global minima shall be considered to tune the ANN model structure.

The selection of the process variables is also a very critical area that will determine the accuracy and viability in modeling the primary reformer TMT. Another reliable statistical analysis can be performed to investigate the relationship between variables to select the most critical variables and to eliminate the redundancy.

The model of primary reformer TMT that has been developed is based on the average TMT of 144 tubes (each chamber) and not the individual tubes. For more accurate and reliable TMT values, individual modeling to predict each of the individual TMT can be developed, but it will consume a lot of time (in modeling) and requires higher memory capacity and processing speed for real-time/simulation application.

More fresh data shall be obtained to provide an extensive database in developing a robust model. The reliability of the data is also a constraint in developing a robust model where by measuring the TMT using pyrometer is not really an accurate and reliable method. This constraint could affect the development of the model and inaccuracy of the prediction.

REFERENCES

[1] S.K. Bhaumik, R.Rangaraju, M.A. Parameswara, T.A. Bhaskaran, M.A. Venkataswamy, A.C. Raghuram, R.V. Krishna, 2002, "Failure of reformer tube of an ammonia plant," *Engineering Failure Analysis*, 9, 553-561

[2] J. Brightling, Mar/Apr 2002, "Managing Steam Reformer Tubes," *ABI/INFORM Trade & Industry, 256,* pg. 29

[3] Anonymous, Mar/Apr 2001, "Primary Reformer Problems," *ABI/INFORM Trade & Industry*, 250, pg. 30

[4] V.N. Patel, June 2009, "Modeling of Primary Reformer Tube Metal Temperature (TMT)", Thesis

[5] V.R Harindran, A. Kajah, August 2008, "Online Predictive Tube Metal Temperature (TMT) Modeling", PowerPoint Presentation, P-APC (PETRONAS Advanced Process Control)

[6] A.K. Ray, S.K. Sinha, Y.N. Tiwari, J. Swaminathan, G. Das, S. Chaudhuri, R. Singh, 2003, "Analysis of failed reformer tubes," *Engineering Failure Analysis,* 10, 3510362

[7] F. tadeo and M. J. Grimble, December 2002, "Advanced control of a hydrogen reformer," *Computing & Control Engineering Journal,* 305-314

[8] I.M Alatiqi, A.M Meziou and G.A Gasmelsees, 1989, "Modelling, simulation and sensitivity analysis of steam methane reformers", *Int. Hydrogen Energy*, 14, 241-256

[9] H. Demuth and M. Beale: "Neural Network Toolbox User's Guide", Natick, USA: The MathWorks Inc., 2009

[10] E. Ogasawara, L. Murta, G. Zimbrao, M. Mattoso, June 14-19 2009, "Neural Networks Cartridges for Data Mining on Time Series", Proceedings of International Joint Conference on Neural Networks

[11] L. Ljung, "System Identification Toolbox User's Guide", 6th ed. Natick, USA: The MathWorks Inc., 2006

[12] M. H. Fazalul Rahiman, M. N. Taib, Y. M. Salleh, June 27-28 2006, "Black Box Modeling Of Steam Temperature", 4th Student Conference on Research and Development 2006

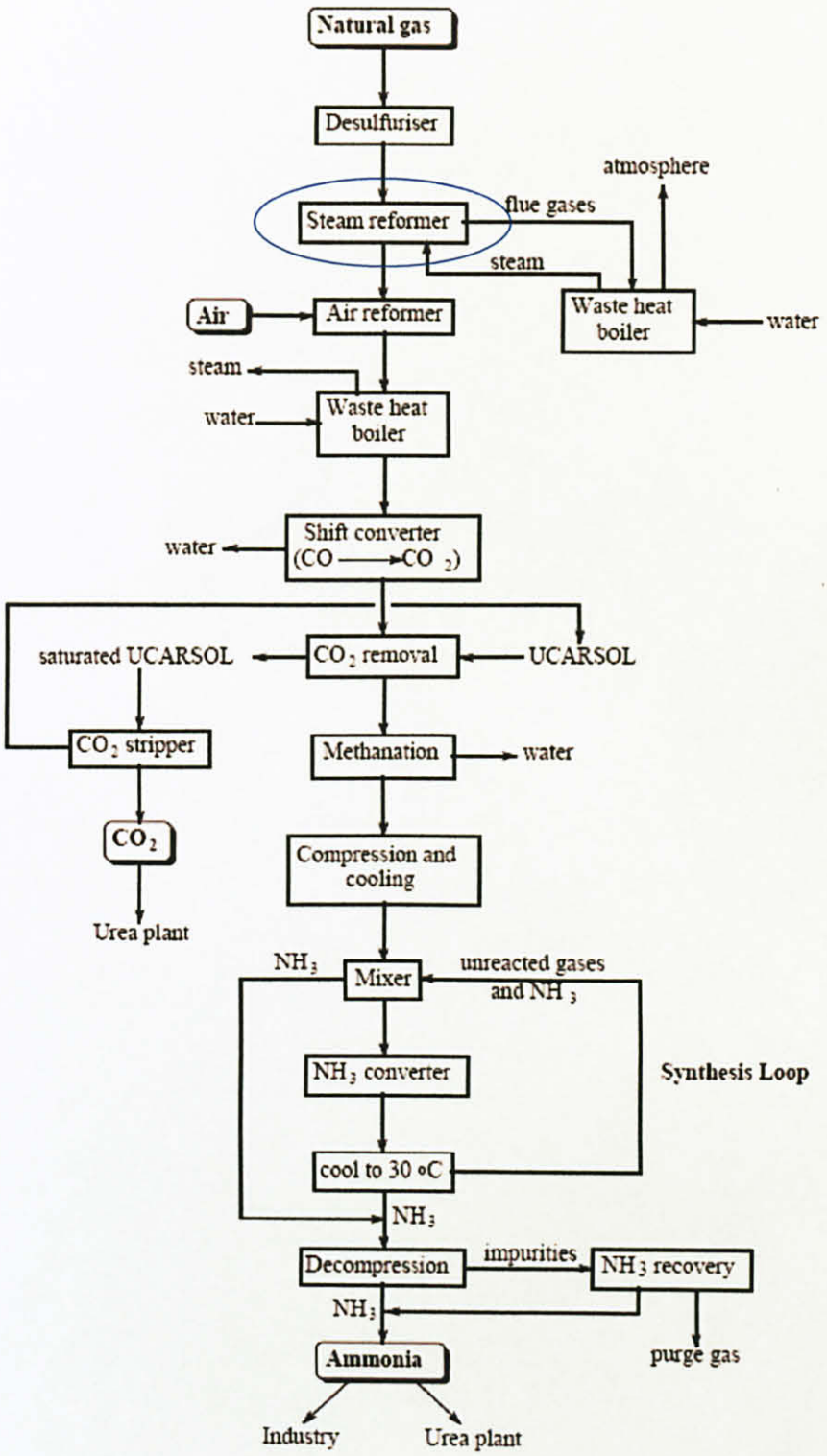
[13] J.C.M. Pires, F.G. Martin, S.I.V. Sousa, M.C.M. Alvim-Ferraz, M.C. Pereira, June 2007, "Selection and Validation of Parameters in Multiple Linear and Principal Component Regressions," Environmental Modeling & Software, 23, 50-53

[14] A. Singh, R.K. Panda and N. Pramanik, September 2009, "Appropriate Data Normalization Range for Daily River Flow Forecasting Using Artificial Neural Network", Water: A vital resource under stress- How Science can help

APPENDICES

Appendix A

SCHEMATIC OF AMMONIA SYNTHESIS PROCESS



Appendix B

PROJECT GANNT CHART FYP I

No.	Details/Week	1	2	3	4	5	6	7	8	9		10	11	12	13	14
1	Selection of Project Topic															
2	Preliminary research work										M					
3	Submission of Preliminary Report										I					
3	Statistical Analysis										D					
4	Model Development										S					
5	Submission of Progress Report										E					
6	Training										M					
7	Seminar 1 (Optional)															
10	Analysis results															
11	Submission of Final Draft Report	**19th October 2009**														
12	Submission of Interim Report	**28th October 2009**														
13	Oral Presentation	**1st – 4th December 2009**														

PROJECT GANNT CHART FYP II

No.	Details/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Planning for FYP II														
2	Literature Review														
3	Develop benchmark models, ARX and MLR development														
4	Neural Network Model Development: Modifying and verifying neuron numbers														
5	ICIAS2010 Paper														
6	Submission of Progress Report I														
7	Project work continues: Neural Network Development														
8	Submission of Progress Report II														
9	Pre-Engineering Design Exhibition (EDX)														
10	Submission of Dissertation (Draft)														
11	Submission of Dissertation (Final Hard Cover)										**25th June 2010**				
12	Oral Presentation										**After exam week**				

MIDSEM

Appendix C

LIST OF PROCESS VARIABLES AND CORRELATION COEFFICIENTS

	Process Variable(s)	Correlation Coefficient(s)		Function(s)	Unit(s)
		Chamber 1	Chamber 2		
X1	FRCA1202	0.795	0.803	Natural Gas Feed	Nm3/h
X2	FRCA1204	0.549	0.541	Total Process Steam	Kg/h
X3	FRCA1304	-0.335	-0.295	Recycle CO2	Nm3/h
X4	TIA1212	0.704	0.711	Feed Temperature	Deg C
X5	PR1210	0.637	0.622	Feed Pressure	Kg G
X6	FIC1250	0.018	-0.026	Natural Gas Fuel	Nm3/h
X7	FIC1254	-0.310	-0.365	Off Gas N2 Wash	Nm3/h
X8	FIC1252	0.405	0.431	Off Gas PSA	Nm3/h
X9	FIC1253	0.514	0.537	Off Gas Cold Box	Nm3/h
X10	FIA1216	0.784	0.779	Combustion Air	Nm3/h
X11	PICA1253	0.679	0.626	Fuel Pressure	Kg G
X12	FFRA1208	-0.791	-0.796	S/C Ratio	-
X13	FICA1251	0.766	0.767	Total Calorific Energy	GJ/h

X14	C1	-0.085	-0.098	C1 Composition	-
X15	TI1220	0.295	0.252	Air Temperature	Deg C
X16	QIA1204	0.390	0.407	Methane Slip	%
X17	TRA1232	0.799	0.773	Reformer Outlet Temperature	Deg C
X18	TRA1234	-0.085	0.569	Flue Gas Temperature	Deg C
X19	TIA1235	0.295	0.659	Flue Gas Temperature	Deg C

Appendix D

MLP OR ELM MATLAB CODING AND VALUES

```
%Latest Program for Neural Network model (MLP or ELM)
%For ELM model, change function to 'newelm'
%This program is for individual modeling, for continuous with
increasing %number, select another programme
%2 layer network (1 hidden layer)
%Normalization from 0.1 to 0.9
% using 199 data for chamber 1 and chamber 2 - OUTLIER REMOVED
%RMSE and FIT as performance indicator

% Clear workspace and command window
clear;
clc;

%load all the input and output
x = load ('TESTdata/199i7try.txt')'; %load the INPUT data 199 data
%y = load ('TESTdata/199i7o1try.txt')'; %load the OUTPUT chamber 1
data
y = load ('TESTdata/199i7o2try.txt')'; %load the OUTPUT chamber 2
data

%get the number of input and number of data
train_data = 100; %number of TRAINING data
validation_data = 99; %number of VALIDATION data
numofvar = size(x,1); %number of input
numofout = size(y,1); %number of input

%dividing the data into x training and y validation data
for m=1:numofvar
    for n=1:train_data
    x_t(m,n)=x(m,n);
    end
end

for m=1:numofvar
    for n=1:validation_data
    x_v(m,n)=x(m,n+train_data);
    end
end

for m=1:numofout
    for n=1:train_data
    y_t(m,n)=y(m,n);
    end
end

for m=1:numofout
    for n=1:validation_data
    y_v(m,n)=y(m,n+train_data);
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%% %NORMALISING INPUTS%%%%%%%%%%%%%%%%%%%%%%%%
%normalise the TRAINING INPUT data 0.1 to 0.9
for row = 1:numofvar
    x_t1(row,:)=( (0.8/max(x_t(row,:)-min(x_t(row,:))
)).*(x_t(row,:)-(min(x_t(row,:)))) +0.1  );
end

%normalise the VALIDATION INPUT data
for row = 1:numofvar
    x_v1(row,:)=((0.8/max(x_v(row,:)-
min(x_v(row,:)))).*(x_v(row,:)-(min(x_v(row,:)))) +0.1 );
end

%normalise the TRAINING OUTPUT data
y_t1=((0.8/(max(y_t)-min(y_t)))*(y_t-min(y_t)) +0.1 );

%normalise the VALIDATION OUTPUT data
y_v1=((0.8/(max(y_t)-min(y_t)))*(y_v-min(y_t)) +0.1);

%minimum and maximum value for the training data after
normalization

t = minmax(x_t1);

%%%%%%%%%%%%%%%%%%%%%%%%%NETWORK SETTINGS%%%%%%%%%%%%%%%%%%%%%%%%%
%for layer 1 (hidden layer) and layer 2 (output layer)

neuron_1 = 4; %number of neurons for layer 1
neuron_2 = 1; %number of neurons for layer 2

%auto prompt trainign function
selectnetwork = input('Select input choice. traincgb default 6 ->
');

%selecting training function based on below
%selectnetwork=2; %for manual input to select training algorithm

%network and parameters
if(selectnetwork==1)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'trainrp');
%choice 1
elseif (selectnetwork==2)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'traingda');
%choice 2
elseif (selectnetwork==3)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'traingdx');
%choice 3
elseif (selectnetwork==4)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'traincgf');
%choice 4
elseif (selectnetwork==5)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'traincgp');
%choice 5
elseif (selectnetwork==6)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'traincgb');
%choice 6
elseif (selectnetwork==7)
```

```
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'trainscg');
%choice 7
elseif (selectnetwork==8)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'trainbfg');
%choice 8
elseif (selectnetwork==9)
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'trainlm');
%choice 9

end

net.trainParam.show = 100; %show cost function every X epochs
net.trainParam.lr = 0.2;
net.performFcn='msereg'; %generalization, but the optimal
performance ratio need to be found
net.trainParam.epochs = 2000;
net.trainParam.goal = 0.01;
net=init(net);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INITIALISING%%%%%%%%%%%%%%%%%%%%%%%%

%set the weights and biases to 0

%set the weights for 1st layer to 0
for m=1:neuron_1
    for n=1:numofvar
    w_1(m,n)=0;
    end
end
net.IW{1,1}=w_1;

%set the weights for 2nd layer to 0
for m=1:neuron_2
    for n=1:neuron_1
    w_2(m,n)=0;
    end
end
net.LW{2,1}=w_2;

%set the bias for 1st layer to 0
for m=1:neuron_1
    b_1(m,1)=0;
end
net.b{1}=b_1;

%set the bias for 2nd layer to 0
for m=1:neuron_2
    b_2(m,1)=0;
end
net.b{2}=b_2;

%checking the weights and biases (make sure all are 0)
net.IW{1,1}; %weights of 1st layer
net.LW{2,1}; %weights of 2nd layer
net.b{1}; %bias of 1st layer
net.b{2}; %bias of 2nd layer
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%TRAIN AND VALIDATE%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%train the network

[net,tr]=train(net,x_t1,y_t1);

%simulate the network with TRAINING data
ytrain=sim(net,x_t1); %simulate the network
test1=mse(ytrain-y_t1);
ytrain1 = (((max(y_t)-min(y_t)))*( (ytrain-0.1)/0.8 ))+min(y_t) ;
%denormalise the output [0.1,0.9]
etrain=y_t-ytrain1; %calculate the different between the actual
and predicted temperature value

%simulate the network with VALIDATION data
yvalid=sim(net,x_v1); %simulate the network
yvalid1 = (((max(y_v)-min(y_v)))*( (yvalid-0.1)/0.8 ))+min(y_v) ;
%denormalise the output [0.1,0.9]

evalid=y_v-yvalid1; %calculate the different between the actual
and predicted temperature value

yvalid1=yvalid1';
ytrain1=ytrain1';

%%%%%%%%%%%%%%%%%%%%%%%%RESULTS PRESENTATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%correlation between outputs and target outputs

[m_train,b_train,r_train]=postreg(ytrain1',y_t); % correlation for
training
[m_validate,b_validate,r_validate]=postreg(yvalid1',y_v);  %
correlation for training

onplot=1; %either to on/off plots

if(onplot==1)

%plot the actual and predicted TMT from VALIDATION data
subplot(2,2,1);
plot (yvalid1','r');
hold on;
plot (y_v,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of NN model for Chamber 1 (Validation Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from
VALIDATION data
subplot(2,2,2);
plot(evalid,'*');
xlabel('Set of Data');
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature
for Chamber 1 (Validation Data)');
grid on;
 %plot the actual and predicted TMT from TRAINING data
```

```
subplot(2,2,3);
plot (ytrain1','r');
hold on;
plot (y_t,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of NN model for Chamber 1 (Training Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from
TRAINING data
subplot(2,2,4);
plot(etrain,'*');
xlabel('set of Data');
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature
for Chamber 1 (Training Data)');
grid on;

ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature
for Chamber 1 (Training Data)');
grid on;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%PERFORMANCE INDICATOR%%%%%%%%%%%%%%%%%%%%

%%RMSE CALCULATION

%error analysis for the TRAINING data
rmse_train = sqrt(mse(etrain)) %mean square error
%index_train = (sum((etrain).^2)/sum((y_t-mean(y_t)).^2))*100
%index value

%error analysis for the VALIDATION data
rmse_valid = sqrt(mse(evalid)) %mean square error
%index_valid = (sum((evalid).^2)/sum((y_v-mean(y_v)).^2))*100
%index value

%r_train
r_validate

%%FIT CALCULATION

fite= (1-norm(etrain)/norm(y_t-mean(y_t)))*100 %norm represent
magnitude y_t actual value
fitv = (1-norm(evalid)/norm(y_v-mean(y_v)))*100
%fit2e= (1-norm(evalid)/norm(output2e-mean(output2e)))*100 %norm
represent magnitude
%fit2v = (1-norm(error2v)/norm(output2v-mean(output2v)))*100
```

```
grid on;

end

%error analysis for the TRAINING data
rmse_train = sqrt(mse(etrain)) %mean square error

%error analysis for the VALIDATION data
rmse_valid = sqrt(mse(evalid)) %mean square error

%index_valid = (sum((evalid).^2)/sum((y_v-mean(y_v)).^2))*100
%index value

%r_train
r_train
r_validate

%FIT ADDITION

fit_train= (1-norm(etrain)/norm(y_t-mean(y_t)))*100 %norm
represent magnitude y_t actual value

fit_validate = (1-norm(evalid)/norm(y_v-mean(y_v)))*100
```

ANN (MLP) Model Weights and Biases Values – Traingdx, 53 Neurons

Chamber 1

Input Weights is a {53x7} matrix

Neuron	FIA 1216	FRCA 1202	FICA 1251	TRA 1232	PICA 1253	FFRA 1208	TIA 1212
1 to 53	0.0972	0.1277	0.1005	0.1677	0.1012	-0.1420	0.0986

Layer Weight is a {1x53} matrix

Neuron (Output)	Node (1 to 53)
Chamber 2	0.0827

Input Bias is a {53x1} matrix

Neuron	Bias
1 to 53	-0.0346

Layer Bias is a {1x1} matrix

Neuron (Output)	Bias
Chamber 2	-1.9776

Chamber 2

Input Weights is a {53x7} matrix

Neuron	FIA 1216	FRCA 1202	FICA 1251	TRA 1232	PICA 1253	FFRA 1208	TIA 1212
1 to 53	0.0904	0.1312	0.0946	0.1491	0.0715	-0.1419	0.0982

Layer Weight is a {1x53} matrix

Neuron (Output)	Node (1 to 53)
Chamber 2	0.0782

Input Bias is a {53x1} matrix

Neuron	Bias
1 to 53	-0.0292

Layer Bias is a {1x1} matrix

Neuron (Output)	Bias
Chamber 2	-1.8030

Appendix E

RBF MATLAB CODING

```
%Latest Program for Neural Network model (RBF Model)
%RBF critically depend on the value of spread constant, change
%Normalization from 0.1 to 0.9
%Uing 199 data for chamber 1 and chamber 2 - OUTLIER REMOVED
%RMSE and FIT as performance indicator

% Clear workspace and command window
clear;
clc;

%load all the input and output
x = load ('TESTdata/199i7try.txt')'; %load the INPUT data 199 data
%y = load ('TESTdata/199i7o1try.txt')'; %load the OUTPUT chamber 1
data ->
y = load ('TESTdata/199i7o2try.txt')'; %load the OUTPUT chamber 2
data ->

%get the number of input and number of data
train_data = 100; %number of TRAINING data
validation_data = 99; %number of VALIDATION data

numofvar = size(x,1); %number of input
numofout = size(y,1); %number of input

%diviing the data into x training and y validation data
for m=1:numofvar
    for n=1:train_data
    x_t(m,n)=x(m,n);
    end
end

for m=1:numofvar
    for n=1:validation_data
    x_v(m,n)=x(m,n+train_data);
    end
end

for m=1:numofout
    for n=1:train_data
    y_t(m,n)=y(m,n);
    end
end

for m=1:numofout
    for n=1:validation_data
    y_v(m,n)=y(m,n+train_data);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%% NORMALISING INPUTS %%%%%%%%%%%%%%%%%%%%%%%%
```

```
H=0.9;
L=0.1;
%normalise the TRAINING INPUT data
for row = 1:numofvar
    x_t1(row,:)=( ((H-L)/max(x_t(row,:)-min(x_t(row,:))
)).*(x_t(row,:)-(min(x_t(row,:)))) +L  );
end

%normalise the VALIDATION INPUT data
for row = 1:numofvar
    x_v1(row,:)=(((H-L)/max(x_t(row,:)-
min(x_t(row,:)))).*(x_v(row,:)-(min(x_t(row,:)))) +L );
end

%normalise the TRAINING OUTPUT data
y_t1=(((H-L)/(max(y_t)-min(y_t)))*(y_t-min(y_t)) +L );

%normalise the VALIDATION OUTPUT data
y_v1=(((H-L)/(max(y_t)-min(y_t)))*(y_v-min(y_t)) +L);

%minimum and maximum value for the training data after
normalization
%(should be 0 and 1)
t = minmax(x_t1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CREATE AND
TRAIN%%%%%%%%%%%%%%%%%%%%%%%%
%spread=2.5 %default
spread = input('spread '); %Auto-prompt at command screen
%newrb creates one neuron at a time
%net=newrbe(x_t1,y_t1,spread); %create and TRAIN using only
training data
%net=newrb(x_t1,y_t1,0.000000000001,0.1);
%net=newrb(x_t1,y_t1,0.00000001,0.075);
net=newrbe(x_t1,y_t1,spread);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VALIDATE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%simulate the network with TRAINING data
ytrain=sim(net,x_t1); %simulate the network
ytrain1 = (((max(y_t)-min(y_t)))*( (ytrain-L)/(H-L) ))+min(y_t) ;
%denormalise the output [0.1,0.9]
etrain=y_t-ytrain1; %calculate the different between the actual
and predicted temperature value

%simulate the network with VALIDATION data
yvalid=sim(net,x_v1); %simulate the network
yvalid1 = (((max(y_t)-min(y_t)))*( (yvalid-L)/(H-L) ))+min(y_t) ;
%denormalise the output [0.1,0.9]
evalid=y_v-yvalid1; %calculate the different between the actual
and predicted temperature value

yvalid1=yvalid1'; %?
ytrain1=ytrain1';
```

```
%%%%%%%%%%%%%%%%%%%%%%%%RESULTS PRESENTATION%%%%%%%%%%%%%%%%%%%%%%%%
%correlation between outputs and target outputs
[m_train,b_train,r_train]=postreg(ytrain1',y_t); % correlation for
training
[m_validate,b_validate,r_validate]=postreg(yvalid1',y_v);  %
correlation for training

onplot=1; %for neurons testing, disable - set to '0'

if(onplot==1)

%plot the actual and predicted TMT from VALIDATION data
subplot(2,2,1);
plot (yvalid1','r');
hold on;
plot (y_v,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of NN model for Chamber 1 (Validation Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from
VALIDATION data
subplot(2,2,2);
plot(evalid,'*');
xlabel('Set of Data');
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature
for Chamber 1 (Validation Data)');
grid on;

%plot the actual and predicted TMT from TRAINING data
subplot(2,2,3);
plot (ytrain1','r');
hold on;
plot (y_t,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of NN model for Chamber 1 (Training Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from
TRAINING data
subplot(2,2,4);
plot(etrain,'*');
xlabel('set of Data');
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature
for Chamber 1 (Training Data)');
grid on;


ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature
for Chamber 1 (Training Data)');
```

Appendix F

ARX MATLAB CODING AND VALUES

```
%ARX410 model

clear
clc;

load io_arx1; %load built ARX model from sysident(GUI)earlier and
input/output

x_enew = detrend(inpute); %de-trending training input data
y_enew = detrend(output1e); %de-trending training output data

x_vnew = detrend(inputv); %de-trending validation input data
y_vnew = detrend(output1v); %%de-trending validation output data 2

x0= inpute-x_enew;
y0= output1e-y_enew;

x1 = inputv-x_vnew;
y1 = output1v-y_vnew;

%simulate the model using training data, for chamber 1 model
ytest = sim(netarx410c1,x_enew);

%simulate the model using validation data, for chamber 1 model
ysim=sim(netarx410c1,x_vnew);

youe=ytest+y0; %trend the output from training model
youv=ysim+y1; %trend the output from validation model

errore=output1e-youe; %calculate error
errorv=output1v-youv;

%%%%%%%%%%%%%%%%%%%%%%%%RESULTS PRESENTATION%%%%%%%%%%%%%%%%%%%%%%%%

fit_estimate = (1-norm(errore)/norm(output1e-mean(output1e)))*100
mser_estimate = mse(errore);
mse_estimate=sqrt(mser_estimate)
index_estimate = (sum((errore).^2)/sum((output1v-
mean(output1v)).^2))*100;

fit_validate = (1-norm(errorv)/norm(output1v-mean(output1v)))*100
mser_validate = mse(errorv);
mse_validate=sqrt(mser_validate)
index_validate = (sum((errorv).^2)/sum((output1v-
mean(output1v)).^2))*100;

subplot(2,2,1);
plot (youe,'r');
hold on;
plot (output1e,'b'); %BLUE colour for actual
```

```
xlabel('Set of Training Data');
ylabel('Tubes Average Temperature (degC)');
title('ARX model for Chamber 1 - Training Set');
legend('Predicted Temperature','Actual Temperature');
grid on;


subplot(2,2,2);
plot(errore,'*')
xlabel('Set of Training Data')
ylabel('Error (degC)');
title('Training Error for Chamber1 - Training Data')
grid on;

subplot(2,2,3);
plot (youv,'r');
hold on;
plot (output1v,'b');
xlabel('Set of Validation Data');
ylabel('Tubes Average Temperature (degC)');
title('ARX model for Chamber 1 - Validate Set');
legend('Predicted Temperature','Actual Temperature');
grid on;

subplot(2,2,4);
plot(errorv,'*')
xlabel('Set of Validation Data')
ylabel('Error (degC)');
title('Validation Error for Chamber1 - Validation Data')
grid on;
```

MATHEMATICAL COEFFICIENTS FOR ARX MODELS

i) ARX 210 Model

na=[2], nb[1 1 1 1 1 1 1], nk=[0 0 0 0 0 0 0]

Sampling interval: 1
Training data input = 100
Validate data input = 99

Chamber 1-ARX210

Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)

Input (7) / output (1)	**Parameter**	**Value(s)**
Chamber 1's TMT (Output)	A(q)	1-0 .3751 q^-1 -0.007163 q^-2
FIA-1216	B1(q)	-0.0004088
FRCA-1202	B2(q)	-0.001145
FICA-1251	B3(q)	0.01187
TRA-1232	B4(q)	0.6766
PICA-1253	B5(q)	167
FFRA-1208	B6(q)	-44.57
TIA-1212	B7(q)	0.201

Chamber 2-ARX210

Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)

Input (7) / output (1)	**Parameter**	**Value(s)**
Chamber 1's TMT (Output)	A(q)	1-0 .4935 q^-1 +0.1182 q^-2
FIA-1216	B1(q)	-0.0004106
FRCA-1202	B2(q)	-0.001206
FICA-1251	B3(q)	7.38e-0.05
TRA-1232	B4(q)	0.7659
PICA-1253	B5(q)	76.57
FFRA-1208	B6(q)	65.47
TIA-1212	B7(q)	0.1871

ii) ARX 410 Model

na=[4], nb[1 1 1 1 1 1 1], nk=[0 0 0 0 0 0 0]

Sampling interval: 1
Training data input = 100
Validate data input = 99

Chamber 1-ARX410

Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)

Input (7) / output (1)	**Parameter**	**Value(s)**
Chamber 1's TMT (Output)	A(q)	1-0 .2624 q^-1 -0.04321q^-2 + 0.01739 q^-3 + 0.02397 q^-4
FIA-1216	B1(q)	-0.0004022
FRCA-1202	B2(q)	0.0003683
FICA-1251	B3(q)	0.0201
TRA-1232	B4(q)	0.7264
PICA-1253	B5(q)	177.8
FFRA-1208	B6(q)	-11.86
TIA-1212	B7(q)	0.0562

Chamber 2-ARX410

Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)

Input (7) / output (1)	**Parameter**	**Value(s)**
Chamber 1's TMT (Output)	A(q)	1-0 .3857 q^-1 -0.1132 q^-2 – 0.1036 q^-3 + 0.067q^-4
FIA-1216	B1(q)	-0.0004532
FRCA-1202	B2(q)	0.0005502
FICA-1251	B3(q)	0.009771
TRA-1232	B4(q)	0.8023
PICA-1253	B5(q)	95.34
FFRA-1208	B6(q)	-29.68
TIA-1212	B7(q)	0.04492

Appendix G

MLR MATLAB CODING AND VALUES

```
%MLR MATLAB CODING

clear;
clc;
load mlr_io; %loading from the workspace earlier saved

%dividing data for each input and output into training and
validation data sets

input1e=input1(1:100);
input2e=input2(1:100);
input3e=input3(1:100);
input4e=input4(1:100);
input5e=input5(1:100);
input6e=input6(1:100);
input7e=input7(1:100);
output1e=output1(1:100);
output2e=output2(1:100);

input1v=input1(101:199);
input2v=input2(101:199);
input3v=input3(101:199);
input4v=input4(101:199);
input5v=input5(101:199);
input6v=input6(101:199);
input7v=input7(101:199);
output1v=output1(101:199);
output2v=output2(101:199);


X1e = [ones(size(input1e)) input1e input2e input3e input4e input5e
input6e input7e];
X1v = [ones(size(input1v)) input1v input2v input3v input4v input5v
input6v input7v];
X2e=X1e; %input for chamber1 = chamber2, training and validation
set
X2v=X1v;

a1 = X1e\output1e
a2 = X1e\output2e


Y1e = X1e*a1;
Y1v = X1v*a1;

Y2e = X2e*a2;
Y2v = X2v*a2;


error1e=output1e-Y1e;
```

```
error1v=output1v-Y1v;
error2e=output2e-Y2e;
error2v=output2v-Y2v;

%%%%%%%%%%%%%%%%%%%%%%%%RESULTS PRESENTATION%%%%%%%%%%%%%%%%%%%%%%%%

rmse1e=sqrt(mse(error1e))
rmse1v=sqrt(mse(error1v))
rmse2e=sqrt(mse(error2e))
rmse2v=sqrt(mse(error2v))

fit1e= (1-norm(error1e)/norm(output1e-mean(output1e)))*100 %norm
represent magnitude
fit1v = (1-norm(error1v)/norm(output1v-mean(output1v)))*100
fit2e= (1-norm(error2e)/norm(output2e-mean(output2e)))*100 %norm
represent magnitude
fit2v = (1-norm(error2v)/norm(output2v-mean(output2v)))*100

subplot(2,2,1);
plot(output1e,'r')
hold on
plot(Y1e,'b')
xlabel('Set of Training Data')
ylabel('Tubes Average Temperature (degC)');
title('Multiple Linear Ression Model for Chamber 1 - Training
Set');
legend('Predicted Temperature','Actual Temperature');
grid on;

subplot(2,2,2);
plot(error1e,'*')
xlabel('Set of Training Data')
ylabel('Error (degC)');
title('Training Error for Chamber1 - Training Data')
grid on;

subplot(2,2,3);
plot(output1v,'r')
hold on
plot(Y1v,'b')
xlabel('Set of Validation Data')
ylabel('Tubes Average Temperature (degC)');
title('Multiple Linear Ression Model for Chamber 1 - Validation
Set');
legend('Predicted Temperature','Actual Temperature');
grid on;

subplot(2,2,4);
plot(error1v,'*')
xlabel('Set of Validation Data')
ylabel('Error (degC)');
title('Validation Error for Chamber1 - Validation Data')
grid on;
```

MATHEMATICAL COEFFICIENTS FOR MLR MODEL

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots \beta_n x_n + \varepsilon$

Chamber 1

β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
462.7084	-0.0001	-0.0013	0.0199	0.6506	163.0549	-56.5246	0.0135

Chamber 2

β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
541.0517	-0.0001	-0.0007	0.0145	0.6219	69.0486	-71.7512	-0.0092

Appendix H

TMT MANUAL DATA ENTRY FORM

REFORMER (F3-1201) TUBES METAL TEMPERATURE SURVEY (TMT)

CHAMBER 1

Tube No.	LEVEL 1					LEVEL 3				
		Row A		Row B	delta T		Row A		Row B	delta T
1N		987		981	6		995		991	4
2N	W	980	W	978	2	W	993	W	990	3
3N		979		980	-1		990		990	0
4N	1026	973	1029	972	1	1069	990	1069	989	1
5N		958		956	2		995		990	5
6		973		977	-4		984		989	-5
7		978		984	-6		989		989	0
8		988		989	-1		989		993	-4
9		993		994	-1		995		995	0
10N	W	992	W	994	-2	W	995	W	995	0
11		994		995	-1		992		994	-2
12N	1025	986	1023	986	0	1122	995	1125	995	0
13N		978		977	1		994		994	0
14		984		989	-5		994		994	0
15N		993		995	-2		994		994	0
16		990		995	-5		991		991	0
17N		994		990	4		995		995	0
18	W	991	W	989	2	W	994	W	995	-1
19N		994		995	-1		993		992	1
20	1030	977	1023	975	2	1111	991	1131	994	-3
21N		976		974	2		995		993	2
22N		987		989	-2		989		985	4
23N		992		990	2		991		988	3

CHAMBER 2

Tube No.	LEVEL 1					LEVEL 3				
		Row C		Row D	delta T		Row C		Row D	delta T
145N		983		980	3		972		969	3
146N	W	972	W	971	1	W	976	W	975	1
147N		971		969	2		982		976	6
148N	1004	969	996	965	4	1141	975	1153	976	-1
149N		957		961	-4		965		968	-3
150N		967		970	-3		974		973	1
151N		974		975	-1		972		969	3
152		979		979	0		967		972	-5
153N		977		982	-5		968		970	-2
154N	W	977	W	979	-2	W	985	W	980	5
155N		974		974	0		972		970	2
156	1002	959	1003	964	-5	1010	968	1012	966	2
157N		954		949	5		965		960	5
158		980		981	-1		964		961	3
159N		971		976	-5		970		973	-3
160		975		976	-1		958		956	2
161N		980		982	-2		979		980	-1
162N	W	978	W	982	-4	W	964	W	966	-2
163N		980		983	-3		969		963	6
164N	1008	974	1012	972	2	1062	977	1075	979	-2
165		969		969	0		970		971	-1
166N		980		986	-6		984		980	4
167N		982		988	-6		986		988	-2